

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS

TEMA:
DESARROLLO DE UNA INTERFAZ CEREBRAL UTILIZANDO
HERRAMIENTAS IOT PARA EL CONTROL DE DISPOSITIVOS ON-OFF
ORIENTADO A PERSONAS CON CAPACIDADES LIMITADAS

AUTORES:
LUIS ALBERTO MONTALVÁN TANDAZO
CHRISTIAN FERNANDO QUINTE CAIZA

TUTOR:
PhD. PILLAJO ANGOS CARLOS GERMÁN

Quito, julio del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Luis Alberto Montalván Tandazo y Christian Fernando Quinte Caiza con documentos de identificación No 172251006-0 y No 172520886-0 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UNA INTERFAZ CEREBRAL UTILIZANDO HERRAMIENTAS IOT PARA EL CONTROL DE DISPOSITIVOS ON-OFF ORIENTADO A PERSONAS CON CAPACIDADES LIMITADAS, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Luis Alberto Montalván Tandazo
Cédula: 172251006-0

Christian Fernando Quinte Caiza
Cédula: 172520886-0

Quito, julio del 2021.

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DESARROLLO DE UNA INTERFAZ CEREBRAL UTILIZANDO HERRAMIENTAS IOT PARA EL CONTROL DE DISPOSITIVOS ON-OFF ORIENTADO A PERSONAS CON CAPACIDADES LIMITADAS realizado por Luis Alberto Montalván Tandazo y Christian Fernando Quinte Caiza obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, julio del 2021

A handwritten signature in blue ink, consisting of a large, stylized 'C' or 'G' shape with several loops and a long horizontal stroke extending to the left.

PhD. Pillajo Angos Carlos Germán

Cédula de identidad: 1709255119

DEDICATORIA

Dedico mi Trabajo de Titulación realizado a Dios ya que fue el motor principal, aquel que me dio fuerzas para seguir adelante cuando más lo necesité y aquel que me ha ayudado a cumplir una meta más en mi vida.

De igual manera lo dedico de todo corazón, amor y aprecio a mis Padres Wilson Quinte y Zaida Caiza por el esfuerzo y sacrificio brindado, por el apoyo incondicional, por ser guías en mi vida y ayudarme siempre a ser una mejor persona, llena de valores y responsabilidades y por último lo dedico a mis hermanos Tatiana y Alex ya que aun en los momentos más difíciles siempre estuvieron presentes mostrándome todo su apoyo.

Christian Quinte

Dedico este Trabajo de Titulación con mucho cariño a mi padre Luis Montalván y a mi madre Alba Tandazo por ser el pilar fundamental en mi vida, por el apoyo y el esfuerzo que han hecho para poder cumplir mis metas, A mi hermana Taid Montalván por ser el regalo más grande que me ha dado la vida.

A mi familia que me han apoyado en los momentos difíciles y brindándome todo su apoyo. A todas las personas que marcaron mi desarrollo personal y profesional.

Luis Montalván

AGRADECIMIENTO

Agradezco a Dios por regalarme la vida, por darme la dicha de disfrutar una meta más junto a mi familia y seres queridos, por ayudarnos en los momentos más difíciles y nunca abandonarnos.

Muy agradecido con mis Abuelitos ya que fueron pilar fundamental para la culminación de mi carrera, de igual manera a mis padres, hermanos y amigos que siempre estuvieron en los momentos más necesitados.

Al Ing. Carlos Pillajo por ser nuestro tutor en este trabajo de titulación ya que su vasto conocimiento en la materia fue un gran aporte en la culminación de este proyecto.

Por último, agradezco a la Universidad Politécnica Salesiana por darme la oportunidad de realizar mis estudios superiores en sus aulas, laboratorios e instalaciones, bajo la tutela del cuerpo de docentes que fue pieza clave en mi preparación para una vida profesional.

Christian Quinte

Agradezco a Dios por darme salud y guiarme, a mis padres y hermana por su apoyo incondicional, consejos y enseñanzas. A mis Tías Mélida Tandazo y Fanny Tandazo por apoyarme económica y emocionalmente para poder culminar con éxito el presente proyecto.

A la Universidad Politécnica Salesiana y a todos los ingenieros que me ayudaron en mi formación profesionalmente. Agradezco al Ingeniero Carlos Pillajo por su guía, apoyo y conocimiento impartido ya que ha sido fundamental para la culminación exitosa del presente proyecto.

Luis Montalván

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR	2
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	3
DEDICATORIA.....	4
AGRADECIMIENTO	5
RESUMEN.....	12
ABSTRACT	13
INTRODUCCIÓN.....	14
CAPÍTULO 1.....	15
1. ANTECEDENTES	15
1.1. Planteamiento del Problema.....	15
1.2. Justificación.....	16
1.3. Objetivos	16
1.3.1 Objetivo General	16
1.3.2 Objetivos Específicos.....	17
1.4. Metodología	17
1.4.1. Metodología Explicativa	17
1.4.2. Metodología Descriptiva.....	17
1.4.3. Metodología Aplicativa.....	18
1.4.4. Metodología Deductiva.....	18
CAPÍTULO 2.....	19
2. FUNDAMENTACIÓN TEÓRICA.....	19
2.1. Electroencefalograma (EEG)	19
2.2. Captación del EEG.....	20
2.3. Sensor Emotiv Insight.....	21
2.4. Raspberry PI 4B	22
2.5. Módulo ESP32	24

2.5.1. Interfaces GPIO.....	26
2.6. Módulo relé	27
2.7. Cloud Computing.....	28
2.7.1. IBM Cloud	28
2.7.2. FireBase.....	30
2.7.3. Node-RED.....	30
2.8. Protocolo MQTT	31
2.9. Protocolo OSC	32
CAPÍTULO 3.....	34
3. DISEÑO E IMPLEMENTACIÓN DEL DISPOSITIVO.....	34
3.1. Generalidades.....	34
3.2. Proceso de funcionamiento del sistema	34
3.3. Implementación Node-Red	35
3.3.1. Instalación Node-Red en Raspberry Pi 4B.....	35
3.3.2. Instalación Protocolo MQTT	37
3.3.3. Implementación Node-Red	38
3.3.3.1. Programación General.....	39
3.3.3.2. Adquisición de Datos UDP	40
3.3.3.3. Dashboard	41
3.3.3.4. Envío de datos a la nube.....	43
3.4. Implementación de Arduino.....	44
3.5. Creación de servidores en la Nube.....	46
3.5.1. Servidor IBM Cloud.....	46
3.5.1. Servidor FireBase.....	49
3.6. Entrenamiento del Sensor Emotiv Insight.....	51
3.8. Implementación Física	53
3.7. Diagrama Esquemático	54
CAPÍTULO 4.....	57
4. PRUEBAS	57

4.1. Generalidades	57
4.2. Tablas de Tiempo de respuestas.....	58
CAPÍTULO 5.....	62
CONCLUSIONES.....	62
RECOMENDACIONES.....	64
BIBLIOGRAFÍA.....	65
ANEXOS.....	67

ÍNDICE DE FIGURAS

Figura 2.1. Sistemas de posicionamiento de los electrodos	21
Figura 2.2. Sensor Emotiv Insight.....	22
Figura 2.3. Raspberry Pi 4B	23
Figura 2.4. Módulo ESP32.....	25
Figura 2.5. Módulo Relé	28
Figura 2.6. Plataforma IBM Cloud	29
Figura 2.7. Node-Red.....	31
Figura 2.8. Protocolo MQTT	32
Figura 3.1. Diagrama de Funcionamiento.....	34
Figura 3. 2. Instalación Node-Red	36
Figura 3. 3. Selección Aplicación Node-Red.....	36
Figura 3. 4. Instalación Finalizada	37
Figura 3. 5. Instalación protocolo MQTT	38
Figura 3. 6. Implementación Node-Red.....	39
Figura 3.7. Diagrama de flujo del Node-Red.....	39
Figura 3.8. Diagrama de flujo de Datos UDP	40
Figura 3.9. Diagrama de flujo del Dashboard.....	41
Figura 3. 10. Dashboard.....	42
Figura 3.11. Diagrama de flujo de datos en la nube.....	43
Figura 3. 12 IDE Arduino con programación	44
Figura 3.13. Diagrama de flujo de adquisición de datos.....	45
Figura 3. 14 Puerto Serial COM3	46
Figura 3. 15. Instalación de Servicio IoT.....	47
Figura 3. 16. Creación de Dispositivo virtual	47
Figura 3. 17 Visualización de sucesos	48
Figura 3. 18. Creación de tarjeta de Visualización	49
Figura 3. 19. Creación de Servicio Realtime	49
Figura 3. 20. Creación de lector de datos.....	50
Figura 3. 21. Visualización de datos	50
Figura 3. 22 Software Emotiv BCI	51
Figura 3. 23. Verificación de contacto de los sensores	52
Figura 3. 24 Entrenamiento de los comandos mentales.....	52

Figura 3. 25. Implementación física del comando Push	53
Figura 3. 26 Ilustración de estado Neutral	54
Figura 3.27. Conexión de un canal de control	54
Figura 3. 28 Diagrama Esquemático, Diseño Proteus.....	55
Figura 3. 29 Planos Smartswitch.....	56
Figura 4. 1. Tiempo de respuesta en cliente.....	57
Figura 4. 2. Tiempo de respuesta en el servidor	57

ÍNDICE DE TABLAS

Tabla 2.1 Tabla de comparativa de Señales Cerebrales	19
Tabla 2.2 Interfaces GPIO.....	27
Tabla 4. 1. Resultados del envío de datos para el comando Push.....	58
Tabla 4. 2. Resultados del envío de datos para el comando Pull	59
Tabla 4. 3. Resultados del envío de datos para el comando Right.....	59
Tabla 4. 4. Resultados del envío de datos para el comando Left	60
Tabla 4. 5. Resultados del envío de datos para el comando Neutral.....	60

RESUMEN

El sistema propuesto describe la viabilidad que tienen los dispositivos comerciales de uso diario para comunicarse con sistemas de interfaz cerebro-computador mediante herramientas IoT. El sensor a usar en el desarrollo de este sistema es la diadema cerebral Emotiv Insight, la cual proporciona la capacidad de reconocer señales emitidas por nuestro cerebro o mediante el reconocimiento de patrones faciales además de realizar enlaces de comunicación mediante protocolos.

El sistema también dispone de una placa ordenador llamada Raspberry PI con ventajas semejantes a las proporcionadas por un computador portátil, esta ventaja permite al sistema utilizar herramientas IoT para establecer una comunicación entre la placa ordenador y servidor en la nube mediante protocolos de comunicación entre Node-Red.

Este sistema va dirigido hacia las personas con capacidades limitadas tanto superiores como inferiores logrando así, un mejor estilo de vida y además abriendo las puertas a nuevas investigación en el uso de esta tecnología aplicándola no solo en dispositivos ON-OFF sino en muchos más.

ABSTRACT

The proposed system describes the feasibility of everyday commercial devices to communicate with brain-computer interface systems using IoT tools. The sensor to be used in the development of this system is the Emotiv Insight brain headband, which provides the ability to recognize signals emitted by our brain or through facial pattern recognition in addition to performing communication links through protocols.

The system also has a computer board called Raspberry PI with advantages similar to those provided by a laptop computer, this advantage allows the system to use IoT tools to establish a communication between the computer board and server in the cloud through communication protocols between network nodes.

This system is aimed at people with limited abilities, both upper and lower, thus achieving a better lifestyle and also opening the door to new research in the use of this technology by applying it not only in ON-OFF devices but in many more.

INTRODUCCIÓN

El diseño del presente proyecto técnico está dividido en 5 capítulos en los cuales se documenta el diseño, constitución y verificación de una interfaz cerebral inalámbrico utilizando herramientas IoT para el control de dispositivos on-off orientado a personas con capacidades limitadas.

El primer capítulo se detalla los antecedentes del proyecto, la problemática a solucionar, la propuesta de solución, los objetivos a ser alcanzados dentro del proyecto y finalmente cuál es la metodología que se usará para la culminación del proyecto.

El segundo capítulo menciona los fundamentos teóricos utilizados en el desarrollo del sistema, en esta sección se describen parámetros técnicos de cada uno de los componentes utilizados en el sistema.

El tercer capítulo detalla el procedimiento de adquisición de datos al software Node-Red, creación de servidores en la nube, instalación de protocolos de comunicación como OSC y MQTT, realización del algoritmo en el IDE de Arduino, el entrenamiento de los comandos mentales del sensor Emotiv Insight y la implementación física.

En el capítulo cuarto se realizan las pruebas mediante envío de datos por cada comando mental para obtener sus tiempos de respuesta entre cliente y servidor, así como un tiempo de respuesta óptimo para definir un rango de tiempo de respuesta.

En el quinto capítulo se redacta las conclusiones finales con respecto a los objetivos planteados para este proyecto y las recomendaciones para un uso adecuado del sistema.

CAPÍTULO 1

1. ANTECEDENTES

1.1. Planteamiento del Problema

Según los datos de la organización mundial de la salud (OMS) doscientas millones de personas experimentan algún tipo de discapacidad física de las cuales en el Ecuador existe aproximadamente medio millón de personas con alguna discapacidad de los cuales cerca del 50% de estas personas tiene una discapacidad física ya sea en sus extremidades superiores e inferiores. (World Health Organization, 2013).

Las nuevas tecnologías IoT tiene como objetivo operar dispositivos de uso doméstico mediante la red, con un control totalmente remoto y seguro para el usuario; pero solo algunas de estas tecnologías toman en cuenta al porcentaje de personas con discapacidad físicas que tienen abierta la posibilidad de usar estas nuevas tecnologías.

Las personas con discapacidad en sus extremidades superiores e inferiores tienen dificultad al manipular objetos cotidianos ya sea por haber sufrido accidentes o también casos donde hubiesen requerido amputación de estas, siendo una gran barrera entre grupo de personas y la tecnología.

Si tomamos como ejemplo a un sujeto que haya sido amputado sus extremidades superiores tenga la necesidad de manipular un interruptor de luz, o un sujeto con discapacidad en sus extremidades inferiores requiera abrir o cerrar un portón eléctrico, son casos claros donde se puede evidenciar el problema.

Es por estas razones que el desarrollo de una Interfaz cerebral utilizando herramientas IoT para el control de dispositivos on-off orientado a personas con capacidades limitadas procura ser una gran ventaja ya que mejorará el estilo de vida de los usuarios.

1.2. Justificación

En el Ecuador de un total de 474.484 personas con alguna discapacidad, el 46.06% de personas tienen una discapacidad física de las cuales el 45.55% de esta tienen una discapacidad menor al 49% de discapacidad(Consejo Nacional para la igualdad de discapacidades, 2021), en conjunto si se toma en cuenta los datos adquiridos en el informe de la Organización Mundial de la Salud en el año 2020 se mostraba alrededor de 250 millones experimentaba algún tipo de discapacidad física lo que dicta desarrollar e implementar en base a las nuevas tecnologías soluciones que permitan a estas personas reincorporarse a la sociedad.

En el caso de la Interfaz cerebral utilizando herramientas IoT para el control de dispositivos on-off para personas con capacidades limitadas presenta ventajas como una mejor opción para mejorar su estilo de vida y brindarles una mejor comodidad con las nuevas tecnologías a implementarse.

Recalcando lo anterior dicho la importancia del proyecto tiene un carácter práctico debido a que este sistema de control de la interfaz cerebral se presenta como una opción válida y competente, además de ofrecer una amplia variedad de aplicaciones fuera del alcance de este proyecto y destacando un precio accesible al mercado.

1.3. Objetivos

1.3.1 Objetivo General

Desarrollar un interfaz cerebral inalámbrico utilizando herramientas IoT para el control de dispositivos on-off orientado a personas con capacidades limitadas.

1.3.2 Objetivos Específicos

- Analizar la información proporcionada por el sensor EEG EMOTIV INSIGHT, mediante el empleo del software del dispositivo para el desarrollo del control ON-OFF.
- Caracterizar los dispositivos electrónicos como son: sensor EEG EMOTIV INSIGHT, el ordenador de placa Raspberry, módulo Esp32, módulo Relé, para la implementación IoT.
- Desarrollar un sistema de comunicación entre clientes y el servidor en la nube, por medio de herramientas IoT para establecer un control de dispositivos On-Off.
- Verificar la comunicación e interpretación de datos mediante pruebas de funcionamiento y control de dispositivos ON-OFF para el monitoreo de tiempos de respuesta entre cliente y servidor.

1.4. Metodología

1.4.1. Metodología Explicativa

Permite que el investigador se familiarice con el tema que se va a examinar, por lo cual se estudiará la información proporcionada por el sensor EEG SENSOR EMOTIV INSIGHT mediante el empleo del software del dispositivo para el desarrollo del control ON-OFF.

1.4.2. Metodología Descriptiva

Se utiliza para describir las particularidades de una población o fenómeno en estudio, esta se empleará en la caracterización de los dispositivos electrónicos mediante la conexión del sensor EGG SENSOR EMOTIV INSIGHT con el ordenador de placa Raspberry, para la transmisión de los datos hacia un servidor en la nube.

1.4.3. Metodología Aplicativa.

Tiene por objeto encontrar una solución a un problema inmediato al que afronta una sociedad o una organización, proponemos un sistema de comunicación entre clientes y servidores en la nube, por medio de una herramienta IoT para establecer un control de dispositivos ON-OFF.

1.4.4. Metodología Deductiva

Es un argumento donde la conclusión se infiere necesariamente de la premisa “buscamos verificar la comunicación e interpretación de datos mediante pruebas de funcionamiento de monitoreo y control para dispositivos ON-OFF”.

CAPÍTULO 2

2. FUNDAMENTACIÓN TEÓRICA

2.1. Electroencefalograma (EEG)

El electroencefalograma (EEG) posibilita a los usuarios divisar de forma óptima la actividad fisiológica realizada por la corteza cerebral en el cerebro. El electroencefalograma capta el diferencial de potencial eléctrico tomado a través de electrodos que se colocan sobre el cuero cabelludo, dichas señales se forman por la actividad neuronal, la amplitud operada por estas señales están en el orden de los micro volts (μV), (Palaniappan, 2014). De igual manera estos potenciales generados se deben a estímulos detallados que son formados durante el EEG, que comúnmente son de tipo auditivo, visual entre otros varios. Como se muestra en la Tabla 2.1 los canales de un EEG se catalogan en función de la frecuencia de las ondas y cada una asociada a ciertos estímulos (Ramadan & Vasilakos, 2016),

Tabla 2.1 Tabla Característica de las Señales Cerebrales

ONDAS	FRECUENCIA (Hz)	PROPIEDADES	ACTIVIDAD MENTAL RELACIONADO
DELTA (Δ)	0.2Hz - 4Hz	Sección frontal en Adultos. Sección posterior en niños. Ondas de alta amplitud.	Ondas lentas relacionadas a la etapa de sueño profundo
THETA (Θ)	4Hz – 8Hz	Sección Frontal Media (Fz a Cz)	Inconciencia, meditación y somnolencia.
ALPHA (α)	8Hz - 12Hz	Región posterior de la cabeza, ambos lados, amplitud alta en el lado dominante	Relajación y concentración.
MU(μ)	8Hz - 13Hz	Corteza sensorial-motora	Indicador de que las neuronas se encuentran trabajando.
BETA(β)	12Hz - 30Hz	Corteza sensorial – motora, entre C3 y C4, distribución simétrica, más evidente en la parte frontal, ondas de amplitud baja.	Estado de alerta, pensativo y concentración activa.

Características de las ondas cerebrales detallando sus propiedades: («Ondas cerebrales», 2019).

2.2. Captación del EEG

Para realizar la captación de la actividad cerebral se han desarrollado cuatro formas básicas:

- Electrodoes superficiales, que se utilizan sobre el cuero cabelludo.
- Electrodoes basales que son implementados en la base del cráneo.
- En cerebro expuesto se utilizan electrodoes quirúrgicos corticales o intracerebrales.
- Electrodoes quirúrgicos intracraneales utilizados en una localización cerebral profunda.

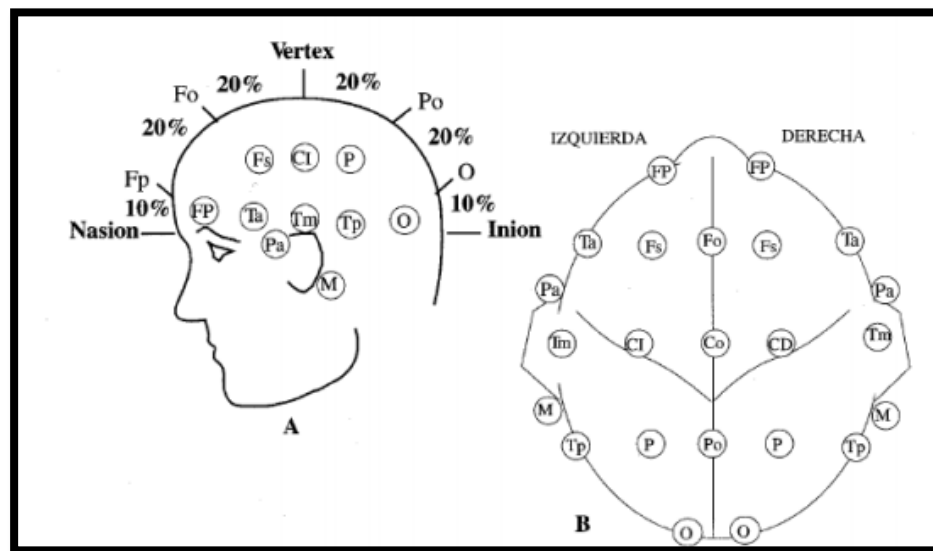
Para el método de captación se emplea tres tipos de registros:

- Electroencefalograma (EEG): Donde se usan electrodoes de superficie o basales.
- Electrocorticograma (ECoG): Usando electrodoes quirúrgicos en la superficie.
- Estéreo Electroencefalograma (E-EEG): se usan electrodoes quirúrgicos de aplicación profunda.

El EEG estándar se utiliza en gran medida en la práctica clínica debido a que es la más práctica ya que no produce dolor, no produce molestia y es accesible para la mayoría de personas. Se realiza colocando un gel conductor con electrodoes superficiales. Se posicionan bajo la norma dictada por el sistema internacional 10-20 como indica la figura 2.1 Cada canal de registro, mide la diferencia de voltaje entre dos electrodoes, lo más habitual es que se usen de 16 a 24 derivaciones en cada montaje.(Ramos-Argüelles et al. 2009).

Los electrodos invasivos permiten estudiar a profundidad áreas cerebrales como la superficie mesial del lóbulo temporal que son de difícil acceso usando tipos como electrodos nasofaríngeos, del foramen oval, esfenoidales, temporales anteriores, etc.

Figura 2.1. Sistemas de posicionamiento de los electrodos



Sistemas internacional 10-20 del posicionamiento de los electrodos. Fuente: (Torres, 2018)

2.3.Sensor Emotiv Insight

El dispositivo Emotiv Insight, brinda la posibilidad de registrar e interpretar las ondas cerebrales en datos que el operador pueda utilizar para establecer una comunicación entre usuarios y otros sistemas como los informáticos, sin la necesidad de usar otros dispositivos, ni una interacción muscular para resolver comandos, además cuenta con un servidor en el cual se almacena toda las lecturas cerebrales hechas por el usuario mediante una cuenta personal en la página. (Villegas, Rojas, 2019).

En este modelo de 5 canales como indica la Figura 2.2 dispone de algoritmos para el reconocimiento de ondas cerebrales mediante patrones faciales y una mejora en el rendimiento en acciones mentales y sus métricas. Incluye como novedad un

giroscopio mediante la detección de movimiento de cabeza siendo esta opción usada como mouse en la mayoría de casos.

Figura 2.2. Sensor Emotiv Insight



Diadema EEG Emotiv Insight de 5 canales. Fuente: (EMOTIV.Inc)

El criterio de selección de este dispositivo es netamente por sus 5 canales de detección para los 5 comandos cerebrales que necesitamos entrenar a nuestro dispositivo para cumplir los objetivos de este proyecto, así como también la facilidad de entrenamiento y comunicación con los demás dispositivos.

2.4. Raspberry PI 4B

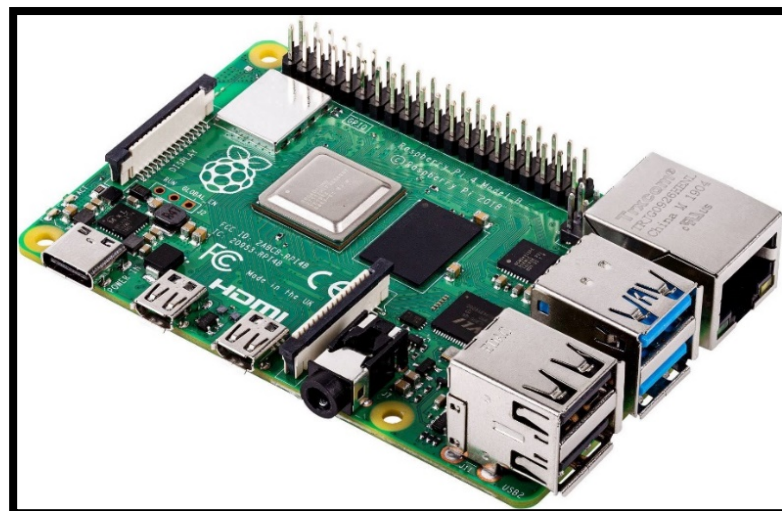
La tarjeta ordenador Raspberry Pi 4B es un mini ordenador de escritorio de tamaño asemejado a una tarjeta de crédito que consta de un bajo costo y además usa un sistema operativo de código abierto, solo se requerirá conectar: un monitor, teclado, ratón y cargar el SO de una tarjeta micro SD. (McGrath, M. 2016).

La Raspberry Pi Foundation proporciona varios sistemas operativos para Raspberry Pi, incluyendo Raspbian y una distribución basada en Debian y como un tercero Ubuntu, Windows 10 IOT Core, RISC OS y distribuciones especializadas para descargar.

Raspberry PI 4B características importantes:

- Microprocesador Quad-core ARMv8 de 1.2 GHz compatible con 64 bits
- 1 GB de Memoria RAM
- Comunicación Bluetooth 4.1 y Wireless N
- Posee interfaces HDMI y cámara
- 4 puertos USB
- Puerto Ethernet
- Puerto para tarjetas microSD
- 40 pines GPIO
- Jacks para audio y video

Figura 2.3. Raspberry Pi 4B



Raspberry Pi 4B. Fuente: (Raspberrypi.org)

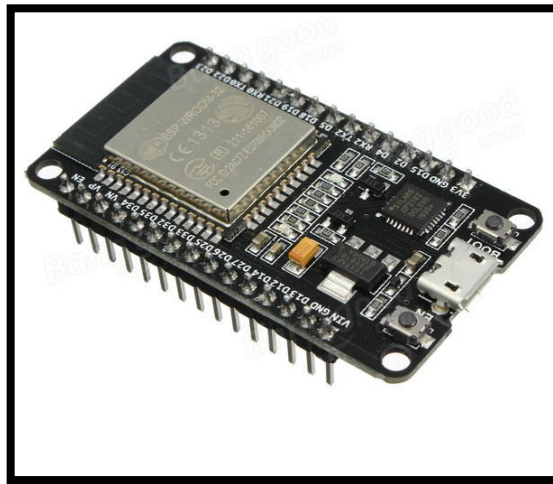
Se utilizó el mini ordenador Raspberry Pi modelo 4B mostrada en la figura 2.3 por la capacidad de procesamiento y la cantidad de memoria RAM ya que utiliza programas en segundo plano, así como también la compatibilidad con el Sensor Emotiv Insight gracias a su librería compatible con el sensor Emotiv Insight.

2.5. Módulo ESP32

El módulo MCU ESP32 es un System On Chip diseñado por Espressif Systems, pero fabricado por TSMC. Se define como un puente para el enlace entre redes o soluciones IoT representando una gran ventaja con respecto a microcontroladores que no disponen conectividad a la red. Además, la serie ESP32 es capaz de ejecutar sus propias aplicaciones de tiempo real, lo que le hace un dispositivo rentable. (Loachamin, 2021).

Las placas ESP32 incorporan un procesador Xtensa LX6 de 2 núcleos como se muestra en la figura 2.4 que le otorga versatilidad para hacer 2 procesos a la vez (Jamal, 2020). Estos núcleos son de bajo coste y además disminuye el consumo de energía, el sistema de comunicaciones inalámbricas incluye tecnologías como amplificadores de bajo ruido y potencia, filtros, y módulos de administración de energía para reducir el consumo de esta cuando se esté usando.

Figura 2.4. Módulo ESP32



Módulo ESP32. Fuente: (Raspberrypi.org)

El criterio de selección de este dispositivo es gracias a su conectividad a redes inalámbricas siendo una solución para utilizar microcontroladores en tecnologías IoT, también teniendo en cuenta el costo en el mercado comparado con otros dispositivos de conectividad a redes inalámbricas.

Especificaciones técnicas principales:

- Procesador de 32 bits tiene una velocidad máxima de 260 MHz.
- Bluetooth: v4.2 BR/EDR y BLE
- Memoria: 520KiB SRAM.
- 4 comunicaciones SPI
- 2 de I2C
- 2 de I2S
- 3 de UART
- Bus CAN 2.0.
- Wifi: 802.11 b/g/n
- 36 pines GPIO; de los cuales 16 son utilizados como salidas de modulación por ancho de pulso (PWM), 18 pines pueden actuar como entradas analógicas.
- Para tratar los puertos analógicos utiliza 2 ADC de entrada múltiple.

- Sus convertidores son de 12 bits, característica que le otorga una mayor resolución para leer las señales analógicas.
- Soporta seguridad estándar de IEEE 802.11
- Cifrado Flash
- Criptografía acelerada por hardware

2.5.1. Interfaces GPIO

El módulo ESP32 posee un total de 25GPIOs como se indica en la tabla 2.2, de los cuales se dividen de la siguiente manera:

- Pines solo de Entrada.
- Pines Pull-up.
- Pines sin enlace interno.
- Pines de entrada analógicas.
- GPIOs de pulsaciones capacitivas.
- Convertidor de digital a analógico (DAC).
- RTC GPIOs
- PWM
- Serie
- I2C
- SPI
- Vcc y GND

Tabla 2.2 Interfaces GPIO

No.	Nombre	Función	Tipo
1	GND	Ground	P
2	3V3	Power supply	P
3	EN	Module-enable signal. Active high.	I
4	SENSOR_VP	GPIO36, ADC1_CH0, RTC_GPIO0	I
5	SENSOR_VN	GPIO39, ADC1_CH3, RTC_GPIO3	I
6	IO34	GPIO34, ADC1_CH6, RTC_GPIO4	I
7	IO35	GPIO35, ADC1_CH7, RTC_GPIO5	I
8	IO32	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9	I/O
9	IO33	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8	I/O
10	IO25	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0	I/O
11	IO26	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1	I/O
12	IO27	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV	I/O
13	IO14	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2	I/O
14	IO12	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3	I/O
15	GND	Ground	P
16	IO13	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER	I/O
17	SHD/SD2*	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD	I/O
18	SWP/SD3*	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD	I/O
19	SCS/CMD*	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS	I/O
20	SCK/CLK*	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS	I/O
21	SDO/SD0*	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS	I/O
22	SDI/SD1*	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS	I/O
34	RXD0	GPIO3, U0RXD, CLK_OUT2	I/O
35	TXD0	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2	I/O
38	GND	Ground	P

Tabla de interfaces GPIO del Módulo ESP32.

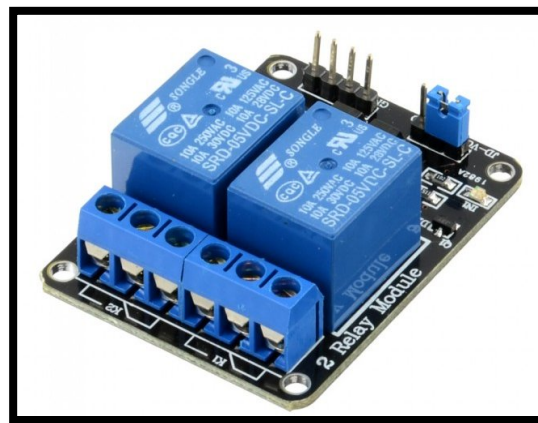
Elaborado por: (Luis Montalván, Fernando Quinte, 2020)

2.6. Módulo relé

El módulo relé posee un transistor para su activación, es decir que no es opto acoplado, posee indicadores LED los cuales se activan dependiendo del estado lógico en la entrada, (rojo representando el suministro de energía y verde para la retransmisión).

Entre algunas características técnicas importantes tenemos que: consta de 3 terminales VCC, GND, una entrada de señal de estado lógico, el módulo relé puede ser activado mediante un microcontrolador, Arduino o Raspberry Pi (Vistrónica, 2020), llegando a manejar cargas máximas de entre 10A hasta 250VAC.

Figura 2.5. Módulo Relé



Módulo Relé. Fuente: (Vistrónica, 2021)

2.7. Cloud Computing

La definición de Cloud Computing o computadora en la nube tiene como característica principal, el cambio en los métodos clásicos en las compañías cuando estas utilizan y reciben los recursos de Tecnología de la Información o TI, obteniendo ventajas como una mayor eficiencia, fácil desarrollo de software, mejoras en la escalabilidad siendo más rápido y masiva. Los nuevos modelos de programación y la nueva infraestructura de TI permitirán que surjan nuevos modelos de negocios.(Murazzo, Millán, 2012).

2.7.1. IBM Cloud

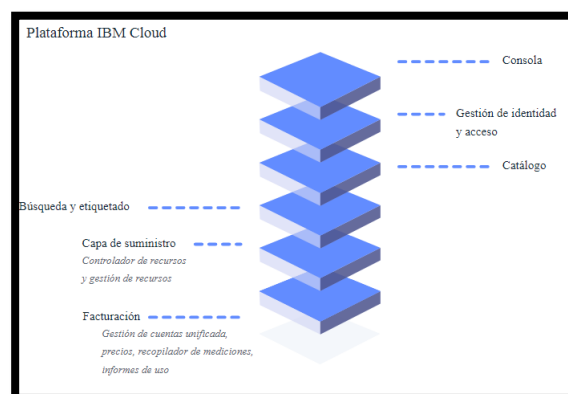
La plataforma de IBM Cloud® consta de la unión de una plataforma como servicio (PaaS) con la infraestructura como servicio (IaaS) logrando una experiencia integrada.

IBM Cloud ofrece soporte desde organizaciones pequeñas como equipos de desarrollo hasta a grandes empresas. IBM es implementada de manera global en centros de datos alrededor de todo el mundo, donde la solución se activa rápidamente y se desempeña de manera segura en un entorno probado y soportado. (Cloud IBM, 2019).

Tal como se muestra en la figura 2.6 la plataforma de IBM Cloud consta de varios componentes que trabajan conjuntamente para proporcionar una experiencia fiable y confiable.

- Una fuerte infraestructura que permita gestionar ver y crear los recursos en la nube, siempre estando en la vanguardia.
- Un servicio que permite tener acceso a los usuarios, de forma segura en la plataforma constando por una gestión de identidad que autentica a los usuarios y controla el acceso a recursos de forma coherente en IBM Cloud.
- Un catálogo que consta de cientos de productos a los que se da soporte
- Un servicio de etiquetado y búsqueda para identificar y filtrar recursos.
- Un mecanismo de facturación y gestión de cuentas que proporciona una utilización exacta de sistema de precios y protección de fraude con tarjeta de crédito.

Figura 2.6. Plataforma IBM Cloud



Composición de la plataforma IBM Cloud. Fuente: (Cloud IBM)

2.7.2. FireBase

FireBase de Google es una plataforma que presta diferentes servicios para ser implementadas en aplicaciones móviles así como también en web. Estos servicios se especializan en proveer a las aplicaciones de funciones en la nube, como almacenamiento online, registros en bases de datos. Al ser una plataforma desarrollada por Google, de la misma forma que Flutter, la integración de los servicios ofrecidos por FireBase es sencilla y precisa de poco tiempo de desarrollo.(Goñi, 2019).

El servicio de la plataforma FireBase almacena su información en la nube usando una base de datos NoSql en formato JSON y adecua esa información con todas las peticiones de la aplicación que estén conectadas. (FireBase, 2018).

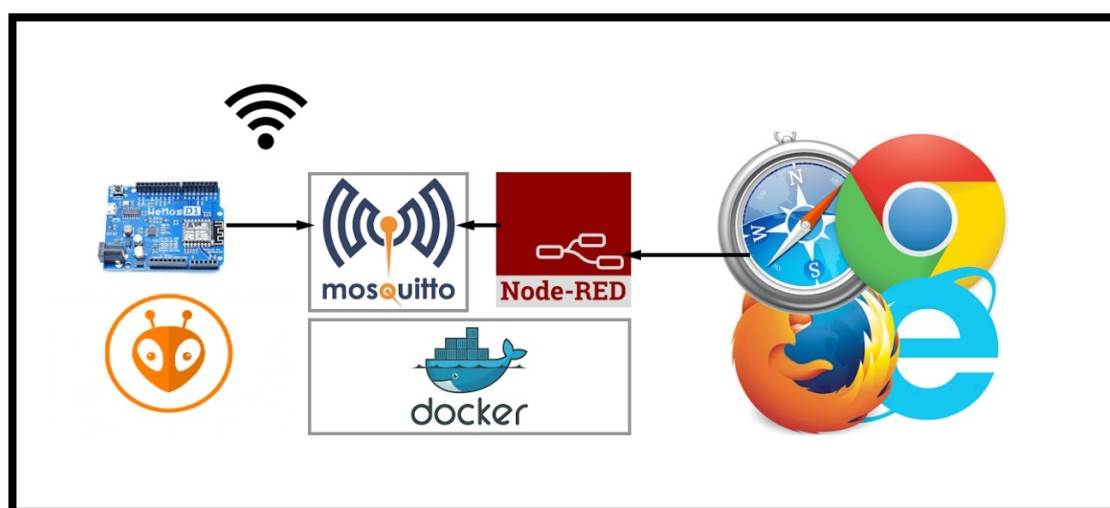
2.7.3. Node-RED

Node-RED proviene de un proyecto abierto, desarrollado por IBM, que propone una herramienta de programación basada en flujos y dirigida por eventos. El comportamiento de la aplicación se representa así como una red de cajas negras, que pueden comunicarse entre sí y regular el flujo de información en el sistema.(Sicari, Rizzardi, and Coen-Porisini 2019).

Esta herramienta basada en una representación visual ayuda a los diseñadores y desarrolladores a comprender las interacciones de control de flujo de datos que existe en toda un red de IoT ya que permite la una real conexión de hardware e interfaces de programación de aplicaciones (API).

Node-Red puede ser instalado en cualquier sistema operativo, además permite la integración de diferentes tecnologías gracias a un amplio catálogo de bibliotecas siendo la más destacada MQTT siendo esta un método ligero de publicación y suscripción.

Figura 2.7. Node-Red



Estructura de Node-Red. Fuente: (Cloud IBM)

2.8. Protocolo MQTT

Message Queue Telemetry Transport (MQTT) es un protocolo de comunicación por encima del protocolo TCP, está basado en el método publish/subscribe. Es un protocolo ligero y fácil de implementar que requiere pocos recursos a nivel de procesamiento y ancho de banda.(Moreno Cerdà 2018).

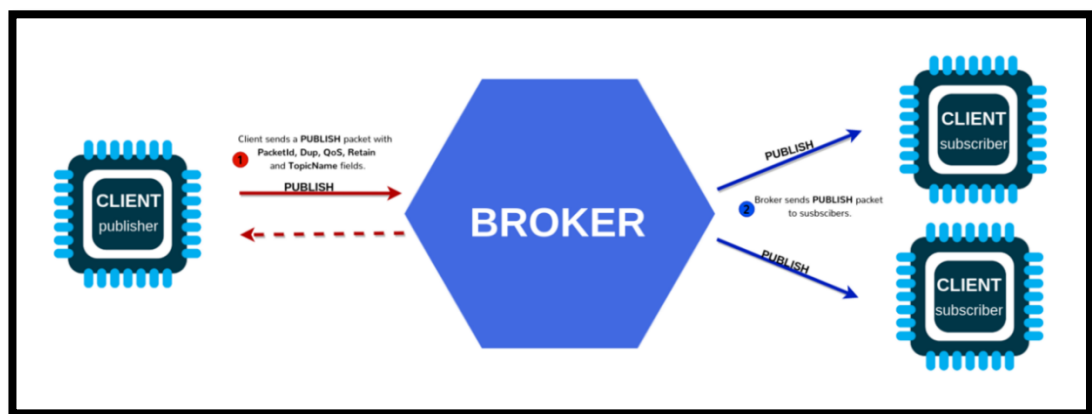
La comunicación entre clientes es asíncrona o near-realtime ya que el protocolo MQTT sigue el modelo publish/subscribe donde los clientes solo son conocedores de la dirección y broker logrando así que los clientes no son conocedores entre unos y otros. El broker cumple con la función de reenviar los mensajes hacia los clientes tomando en cuenta la suscripción mediante tópicos tipo String UTF-8.

Los tópicos deben tener en su estructura un carácter como mínimo y pueden o no disponer de un reconocimiento de niveles que son separados por “/” dando así una estructura jerárquica a los tópicos. Un ejemplo podría ser: “miCasa/primerPiso/luz”.

El protocolo MQTT implementa calidades de servicios (QoS):

- QoS 0 fire and forget, el mensaje es enviado una sola vez y además no realiza ninguna comprobación.
- QoS 1 at least one, requiere un reconocimiento, el mensaje puede ser enviado más de una vez.
- QoS 2 exactly one, hay un sistema de comprobación, se entrega el mensaje una sola vez sin repeticiones.

Figura 2.8. Protocolo MQTT



Estructura del Protocolo MQTT. Fuente: (Issac, 2019)

2.9. Protocolo OSC

OpenSoundControl (OSC) es un codificador de datos para la comunicación de mensajes en tiempo real entre aplicaciones y hardware. OSC fue diseñado originalmente como un método de comunicación altamente preciso, de baja latencia, ligero y flexible para su uso en la interpretación musical en tiempo real. En la actualidad el protocolo OSC ha tenido muchos desarrollos entre ellas como librería, para muchos lenguajes de programación. OSC está situado en los altos niveles de los mejores protocolos, designado en un modelo cliente/servidor y su transferencia de datos se agrupan en una unidad llamada paquete. (Blanco, Francisco, and Edgar 2015).

La unidad de datos básica se llama mensaje con la estructura siguiente:

- Dirección patrón (address pattern). Es una sucesión o eventos OSC que encabeza con el carácter “/”.
- Tipo de cadena (Type tag string). Es una sucesión OSC que encabeza con el carácter “,” los tipos de sucesiones más utilizados son:” i” que representa un entero de 32 bits y “f” que representa un valor flotante.

Características de OSC.

- Esquema de nombre simbólicos tipo URL.
- Expansible
- Dinámico
- Datos de alta resolución de tipo numérico y simbólico.
- Pattern matching que permite la comunicación de varios dispositivos mediante un solo mensaje.
- (time tags) indicadores de tiempo.
- Empaquetamiento de varios mensajes para eventos que deban ocurrir simultáneamente.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN DEL DISPOSITIVO

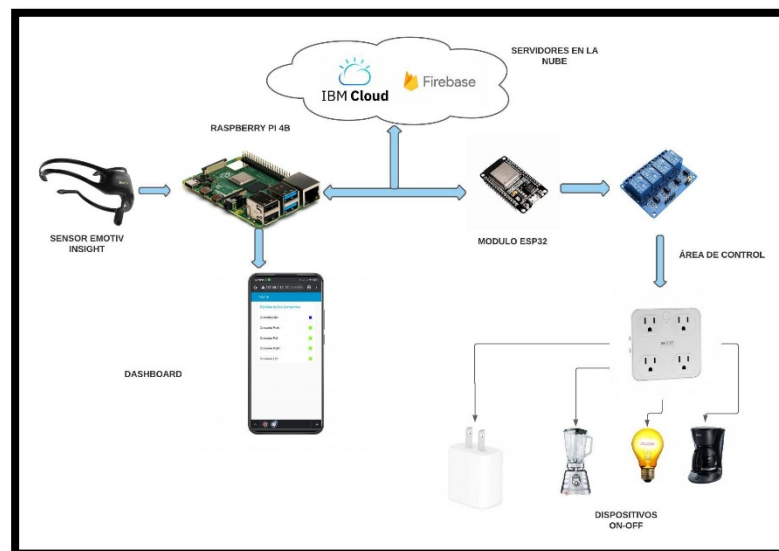
3.1. Generalidades

El sistema desarrollado interpreta los datos emitidos por el sensor Emotiv Insight como un método de control de dispositivos ON-OFF. Los datos que son adquiridos por la diadema Emotiv Insight son señales eléctricas generadas por el cerebro ante la reacción de un estímulo enviándolos a un servidor en la nube. Para este caso se fundamenta el uso de estímulos cerebrales generando comandos mentales los cuales servirán para el control ON-OFF.

3.2. Proceso de funcionamiento del sistema

En la figura 3.1 se ilustra los elementos que se utilizan en el sistema, el sensor Emotiv Insight se encarga de enviar señales eléctricas del cerebro en forma de comandos mentales la cual se enlaza mediante protocolos de comunicación a la placa ordenador Raspberry.

Figura 3.1. Diagrama de Funcionamiento



Elaborado por: Luis Montalván, Fernando Quinte

La placa ordenador Raspberry Pi 4B se encarga de interpretar todas las señales eléctricas emitidas por el sensor Emotiv Insight para luego enviar estos datos mediante protocolo UDP hacia el Node-Red donde son decodificados mediante protocolo OSC. El servidor en la nube recibe los datos decodificados por el Node-Red para posteriormente ser enviados mediante protocolo MQTT hacia el módulo ESP32 conectado mediante una conexión local.

El módulo ESP32 recibe los datos enviados por el Node-Red mediante protocolo MQTT y activa las salidas programadas dependiendo del comando mental enviado hacia la placa ordenador Raspberry donde será este será registrado mediante un mensaje cuando sea activado.

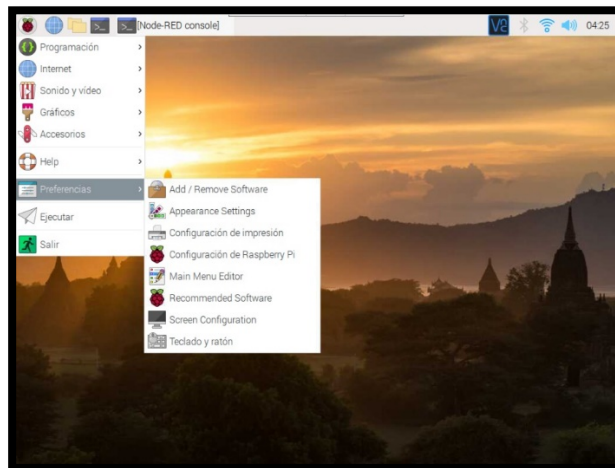
El área de control se encarga de realizar la conversión del voltaje suministrado por el ESP32 de 3.3v a 5v que es el voltaje óptimo para activar el modulo relé y así poder enviar las señales de activación suministradas por los comandos mentales hacia los tomacorrientes.

3.3. Implementación Node-Red

3.3.1. Instalación Node-Red en Raspberry Pi 4B

- Para realizar la instalación de Node-Red en Raspberry nos dirigimos a menú de aplicaciones, preferencias y seleccionamos software recomendados.

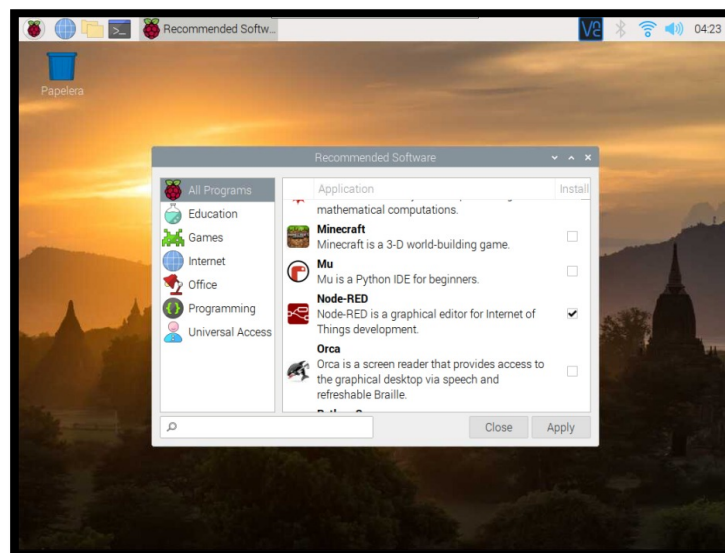
Figura 3. 2. Instalación Node-Red



Elaborado por: Luis Montalván, Fernando Quinte

- Buscamos la aplicación de Node-Red, la seleccionamos y damos en aplicar.

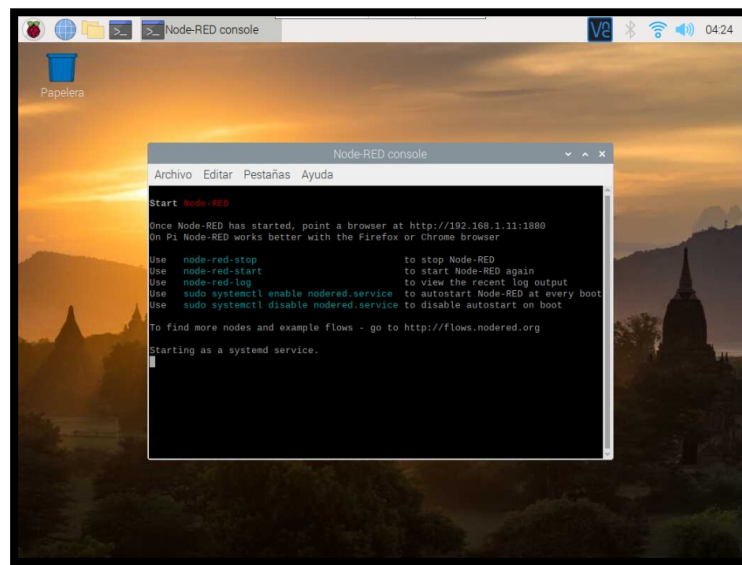
Figura 3. 3. Selección Aplicación Node-Red



Elaborado por: Luis Montalván, Fernando Quinte

- Esperamos que el programa termine su intalación y lo ejecutamos, sera necesario configurarlo para que este se inicie conjunto con la raspberry.

Figura 3. 4. Instalación Finalizada



Elaborado por: Luis Montalván, Fernando Quinte

3.3.2. Instalación Protocolo MQTT

Mosquitto es un servidor local con protocolo MQTT, ya que trabaja desde un nivel de calidad de servicio 0 y posibilita una programación en lenguaje C, esto se lleva a cabo en la placa ordenador Raspberry donde se realiza la instalación de un broker que previamente debe tener instalado el sistema operativo Raspbian mediante los comandos mencionados a continuación

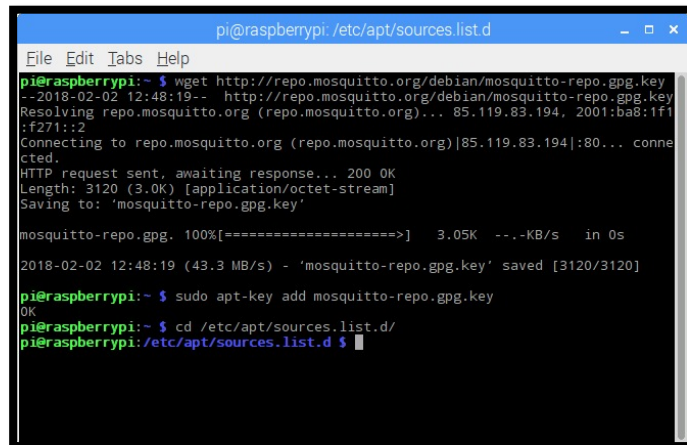
- Descargar la clave para el servidor:
`sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key.`
- Agregar una lista a la clave
`sudo apt-key add mosquitto-repo.gpg.key`
- Ingresar al directorio siguiente:
`cd /etc/apt/sources.list.d/`
- Descargar la lista de repositorios Mosquitto.
`sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list`
- Se actualiza la lista de paquetes mediante
`sudo apt-get update`
- Instalar bróker Mosquitto

Apt-get install mosquitto

- Instalar clientes MQTT

Apt-get install mosquitto-clients

Figura 3. 5. Instalación protocolo MQTT



```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
pi@raspberrypi:~$ wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
--2018-02-02 12:48:19-- http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
Resolving repo.mosquitto.org (repo.mosquitto.org)... 85.119.83.194, 2001:ba8:1f1:f271::2
Connecting to repo.mosquitto.org (repo.mosquitto.org)[85.119.83.194]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3120 (3.0K) [application/octet-stream]
Saving to: 'mosquitto-repo.gpg.key'

mosquitto-repo.gpg. 100%[=====] 3.05K --KB/s in 0s
2018-02-02 12:48:19 (43.3 MB/s) - 'mosquitto-repo.gpg.key' saved [3120/3120]

pi@raspberrypi:~$ sudo apt-key add mosquitto-repo.gpg.key
OK
pi@raspberrypi:~$ cd /etc/apt/sources.list.d/
pi@raspberrypi:/etc/apt/sources.list.d$
```

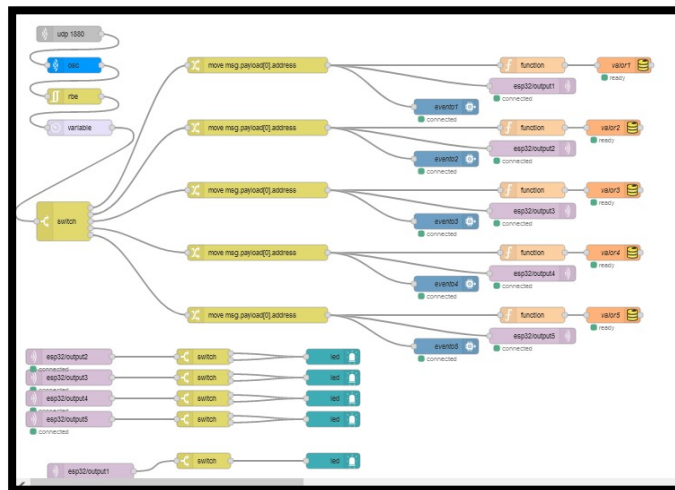
Elaborado por: Luis Montalván, Fernando Quinte

3.3.3. Implementación Node-Red

La implementación de la programación en Node-Red cumple cuatro etapas siendo estas:

- Programación General
- Adquisición de datos UDP
- Dashboard
- Envío de datos a la nube

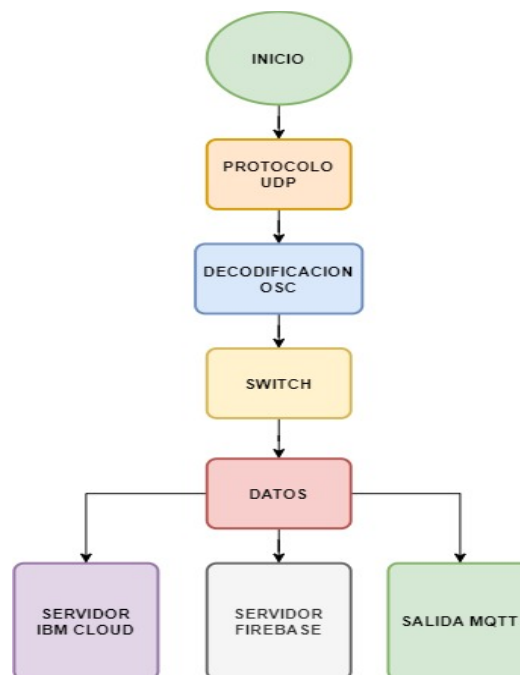
Figura 3. 6. Implementación Node-Red



Elaborado por: Luis Montalván, Fernando Quinte

3.3.3.1. Programación General

Figura 3.7. Diagrama de flujo del Node-Red



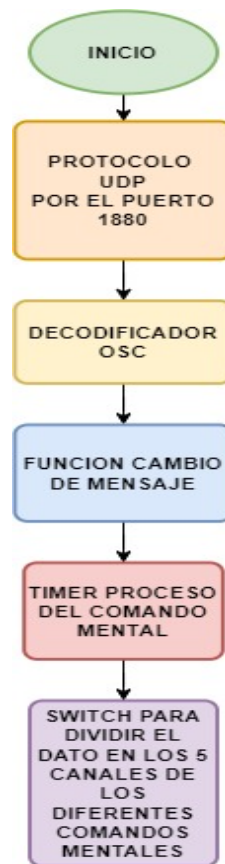
Elaborado por: Luis Montalván, Fernando Quinte

En la figura 3.7 se detalla el proceso que cumple el Node-Red adquiriendo los datos proporcionados por el sensor Emotiv Insight mediante protocolo UDP, siendo estos

decodificados mediante un protocolo OSC, estos datos se envían a un switch el cual cumple la función de distribuirlos hacia los 5 canales de los comando mentales, estos datos se envían hacia el servidor de IBM Cloud, Servidor FireBase y a sus diferentes salidas mediante protocolo MQTT.

3.3.3.2. Adquisición de Datos UDP

Figura 3.8. Diagrama de flujo de Datos UDP



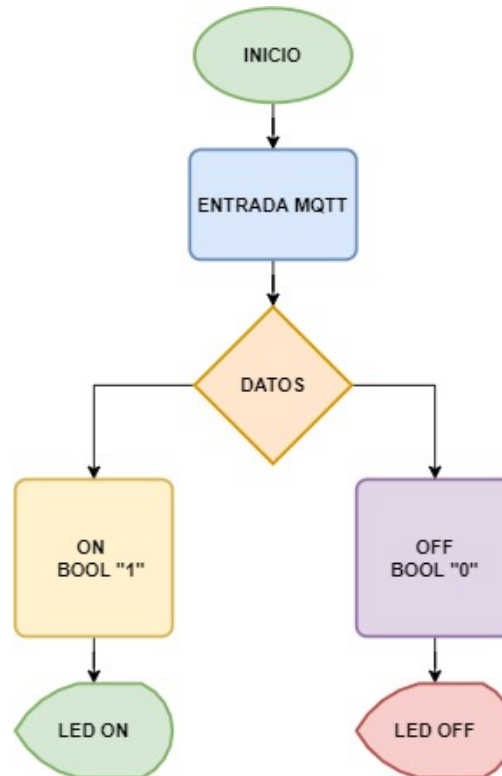
Elaborado por: Luis Montalván, Fernando Quinte

Como se muestra en la figura 3.8 el dato es enviado desde el sensor Emotiv Insight hacia el Node-Red mediante protocolo UDP en el puerto 1880, este dato se decodifica mediante el protocolo Open Sound Control (OSC) el cual entra a una función de cambios de estado de mensaje que los interpretará solo cuando un comando mental sea sustituido por otro, para que un comando mental sea interpretado de una manera correcta este debe cumplir un rango de tiempo óptimo.

Este mensaje es enviado a un switch el cual lo distribuye entre los distintos comandos mentales tales como: Push, Pull, Right, Left y Neutral.

3.3.3.3. Dashboard

Figura 3.9. Diagrama de flujo del Dashboard

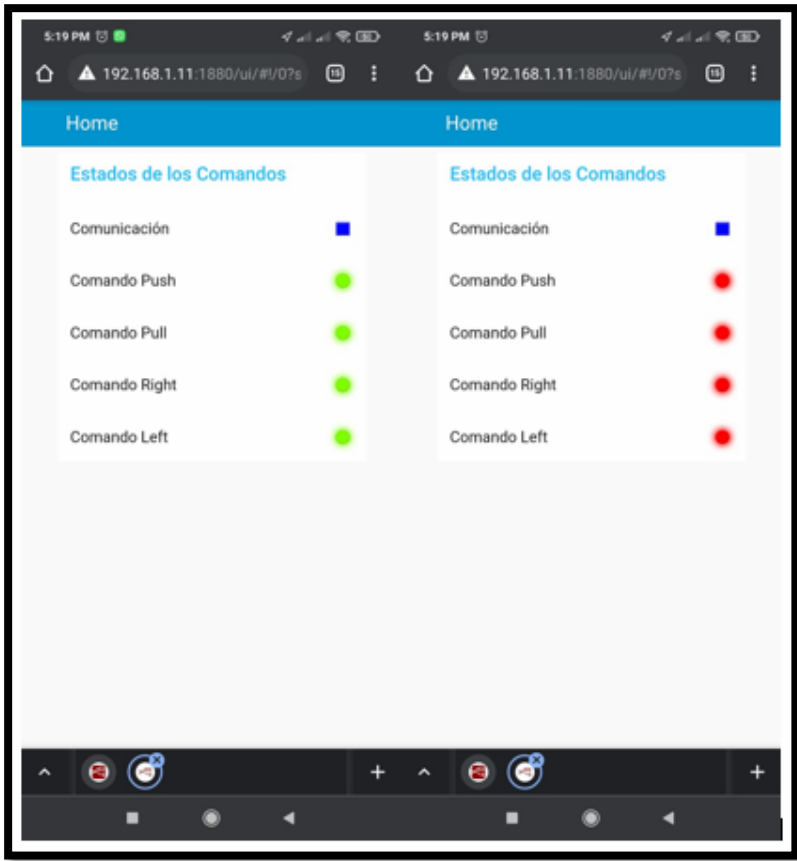


Elaborado por: Luis Montalván, Fernando Quinte

En la figura 3.9 se detalla el proceso de visualización en un Dashboard, el dato es receptado mediante protocolo MQTT llegando a tener dos estados booleanos 0 y 1 (OFF-ON) dando dos instrucciones.

Cuando el dato booleano sea “1” el led indicador del Dashboard cambiará a verde.
Cuando el dato booleano sea “0” el led indicador del Dashboard cambiará a rojo, como se indica en la figura 3.10.

Figura 3. 10. Dashboard



Elaborado por: Luis Montalván, Fernando Quinte

3.3.3.4. Envío de datos a la nube

Figura 3.11. Diagrama de flujo de datos en la nube



Elaborado por: Luis Montalván, Fernando Quinte

En la figura 3.11 se detalla el procesamiento de datos proporcionado por el Node-Red para ser enviados hacia los distintos servidores en la nube, los datos que son enviados al Node-Red son procesados para su correcta interpretación.

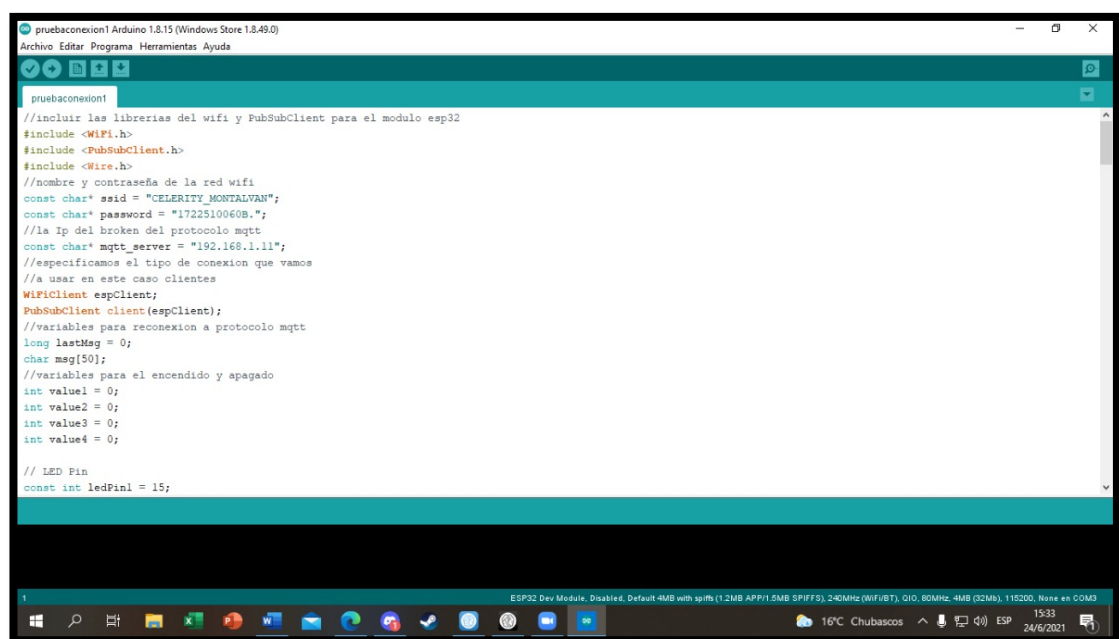
Los datos son enviados a una función donde se autentican con el servidor IBM Cloud y FireBase una vez hecho esta tarea los servidores realizan una verificación de

comunicación entre cliente y servidor, el Servidor lee los datos en su registro de actividades y estos son mostrados en sus diferentes plataformas de visualización

3.4. Implementación de Arduino

Mediante el software IDE de Arduino se realiza el algoritmo de adquisición y transmisión de datos como se muestra en la figura 3.12 para la activación de los actuadores siendo en este caso el modulo Relé mediante comandos mentales proporcionados por los canales de salida del módulo ESP32.

Figura 3. 12 IDE Arduino con programación

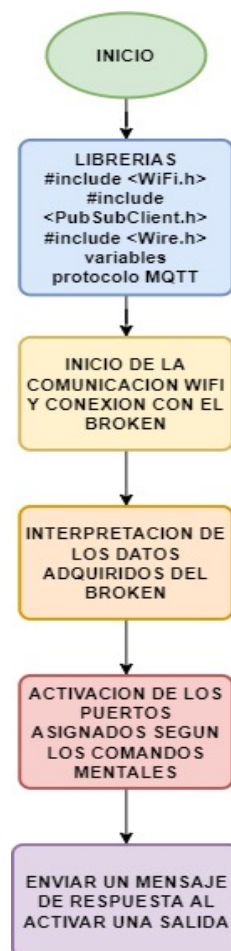


Elaborado por: Luis Montalván, Fernando Quinte

El algoritmo desarrollado para la adquisición y transmisión de datos como se muestra en la figura 3.13 empieza con la declaración de las librerías “WiFi.h”, “PubSubClient.h”, “Wire.h” y las variables a utilizar.

Se realiza una comunicación vía WiFi y con el broker mediante el puerto 1883 además se realiza una verificación de comunicación, la interpretación de datos se realiza mediante tópicos que posteriormente servirán para la activación de los puertos asignados mediante comandos mentales Push, Pull, Right, Left y Neutral, estos comandos mentales enviaran un mensaje de salida cada vez que sean enviados.

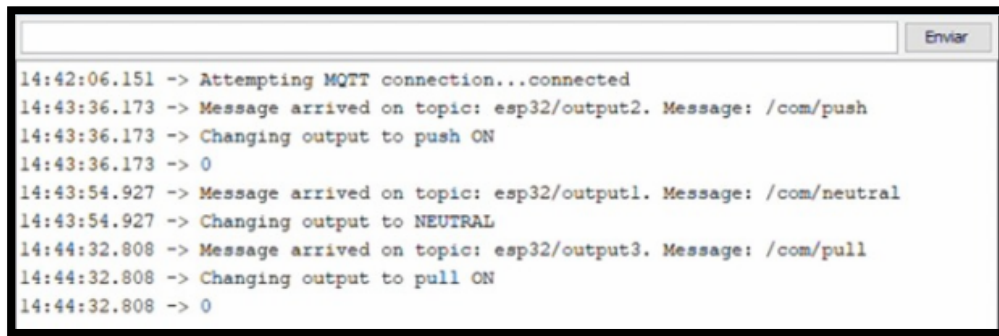
Figura 3.13. Diagrama de flujo de adquisición de datos



Elaborado por: Luis Montalván, Fernando Quinte.

Finalizado la comunicación entre los distintos dispositivos sensor Emotiv Insight, Raspberry, servidor en la nube y ESP32 se visualizará en el puerto Serial COM3 como se indica en la figura 3.14 un mensaje indicando el correcto envío y recepción de datos en cualquiera de los cinco canales que son activados por lo comandos mentales Push, Pull, Right, Left.

Figura 3. 14 Puerto Serial COM3

A screenshot of a serial terminal window titled 'Puerto Serial COM3'. The window has a text area with log messages and a button labeled 'Enviar' in the top right corner. The log messages are as follows:

```
14:42:06.151 -> Attempting MQTT connection...connected
14:43:36.173 -> Message arrived on topic: esp32/output2. Message: /com/push
14:43:36.173 -> Changing output to push ON
14:43:36.173 -> 0
14:43:54.927 -> Message arrived on topic: esp32/output1. Message: /com/neutral
14:43:54.927 -> Changing output to NEUTRAL
14:44:32.808 -> Message arrived on topic: esp32/output3. Message: /com/pull
14:44:32.808 -> Changing output to pull ON
14:44:32.808 -> 0
```

Elaborado por: Luis Montalván, Fernando Quinte.

3.5. Creación de servidores en la Nube

En este apartado se detalla la creación de servidores en la nube proporcionados por IBM Cloud y FireBase los cuales muestran los comandos mentales enviados desde el sensor Emotiv Insight en tiempo real.

3.5.1. Servidor IBM Cloud

- Para poder crear un servidor de IBM Cloud es necesario registrarnos en la página oficial de IBM proporcionada en el siguiente enlace:
<https://cloud.ibm.com/login>.
- Una vez creada la cuenta de IBM se nos desplegará nuestro Área de trabajo y en su buscador escribiremos servicio Internet of Things para poder instalarlo en nuestro servidor en la nube como se indica en la figura 3.15.

Figura 3. 15. Instalación de Servicio IoT

The screenshot shows the IBM Cloud 'Internet of Things Platform' installation page. The main content area has two tabs: 'Crear' (Create) and 'Acerca de' (About). Under 'Crear', there are two sections: 'Seleccione una ubicación' (Select a location) with a dropdown menu showing 'Dallas (us-south)', and 'Seleccione un plan de precios' (Select a pricing plan). Below this is a table with three columns: 'Plan', 'Características' (Features), and 'Tarifas' (Pricing). The 'Lite' plan is selected, showing features like 500 registered devices and 200 MB of data metrics, and a 'Gratuito' (Free) price. On the right, a 'Resumen' (Summary) panel lists the plan details: 'Internet of Things Platform', 'Ubicación: Dallas', 'Plan: Lite', 'Nombre de servicio: Internet of Things Platform-h2', and 'Grupo de recursos: Default'. At the bottom right, there are buttons for 'Crear' (Create), 'Añadir a estimación' (Add to estimate), and 'Ver condiciones' (View conditions).

Plan	Características	Tarifas
Lite	Incluye un máximo de 500 dispositivos registrados y un máximo de 200 MB de métricas de datos cada uno Un máximo de 500 dispositivos registrados Máximo de 500 enlaces de aplicación Un máximo de 200 MB de datos intercambiados, datos analizados	Gratuito

Elaborado por: Luis Montalván, Fernando Quinte.

- Una vez instalado el servicio de Internet of Things proseguimos a realizar la creación de nuestros dispositivos virtuales que se usaran para leer datos en nuestro servidor en la nube mostrado en la figura 3.16, solo bastará con poner un nombre a nuestro dispositivo y un ID.

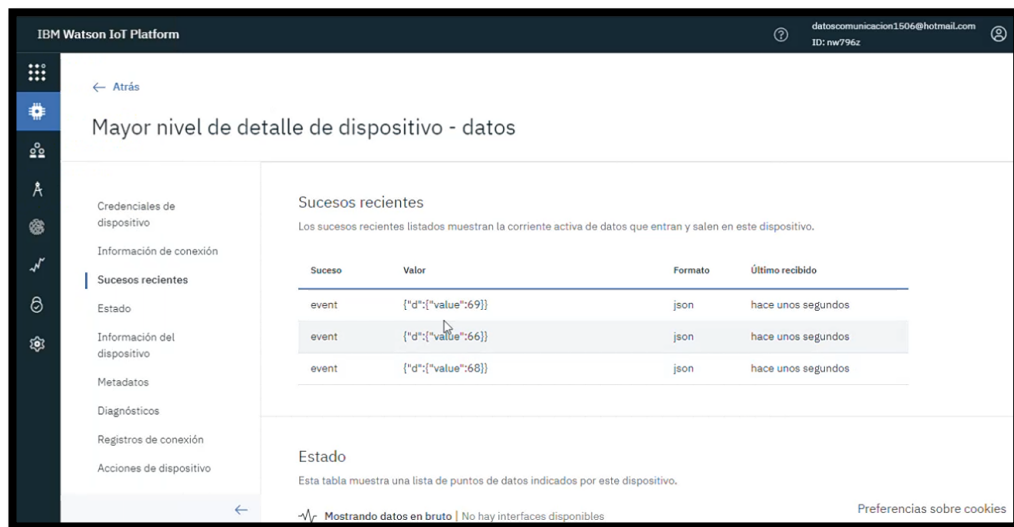
Figura 3. 16. Creación de Dispositivo virtual

The screenshot shows the 'Añadir dispositivo' (Add device) form in the IBM Watson IoT Platform. The form is titled 'Añadir dispositivo' and has a progress bar with four steps: 'Identidad' (Identity), 'Información del dispositivo' (Device information), 'Seguridad' (Security), and 'Resumen' (Summary). The 'Identidad' step is currently active. Below the progress bar, there is a text prompt: 'Seleccione un tipo de dispositivo para el dispositivo que está añadiendo y dé al dispositivo un ID exclusivo.' (Select a device type for the device you are adding and give the device a unique ID). There are two input fields: 'Tipo de dispositivo' (Device type) with the value 'comunicacion' and 'ID de dispositivo' (Device ID) with the value 'dato'. At the bottom right, there are buttons for 'Cancelar' (Cancel) and 'Siguiente' (Next). The top of the page shows the user's email 'deloscomunicacion1506@hotmail.com' and ID 'ID: nw796z'.

Elaborado por: Luis Montalván, Fernando Quinte

- Una vez creado nuestro dispositivo virtual aparecerá en el apartado de dispositivos, este nos desplegará la información necesaria como nombre de la organización, tipo de dispositivo, ID, método de autenticación la cual nos servirán para ingresar en el pallet de Node-Red. Colocamos todos los datos y nos conectamos al servidor y una vez conectado podemos leer cualquier dato que llegue en nuestros sucesos recientes indicado en la figura 3.17.

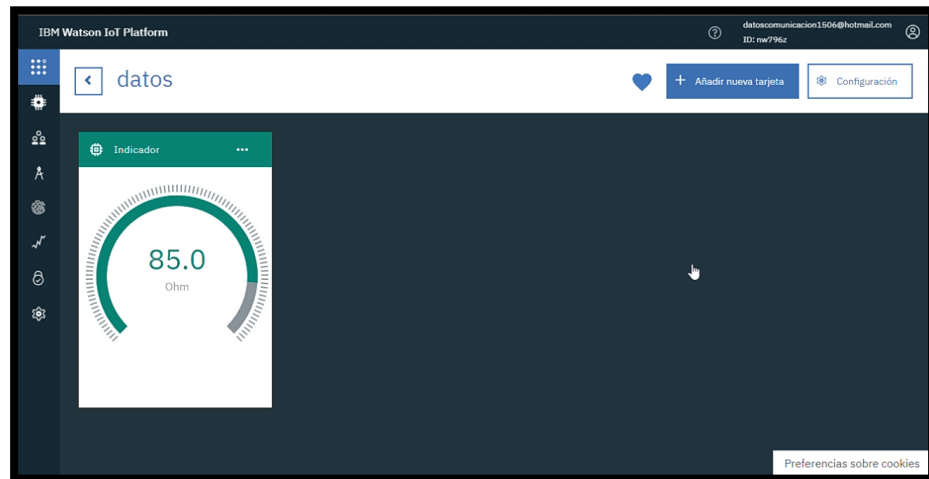
Figura 3. 17 Visualización de sucesos



Elaborado por: Luis Montalván, Fernando Quinte

- En la figura 3.18 se muestra la creación de cartas de visualización para una mejor visualización de los eventos en tiempo real, para crearla no dirigimos al apartado de Paneles indicando el dispositivo e ID del cual queremos visualizar los eventos y posteriormente configuraremos los parámetros a ser mostrados.

Figura 3. 18. Creación de tarjeta de Visualización

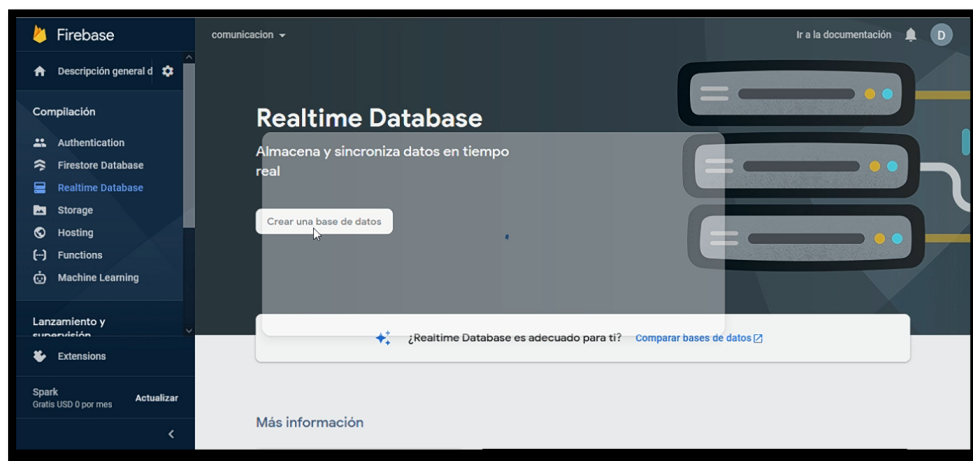


Elaborado por: Luis Montalván, Fernando Quinte

3.5.1. Servidor FireBase

- Para la creación del servidor FireBase nos dirigimos a la página de www.FireBase.google.com la cual nos registramos con un correo electrónico Gmail.
- Una vez registrados crearemos un nuevo proyecto con un servicio Realtime Data para poder leer datos en tiempo real mostrado en la figura 3.19.

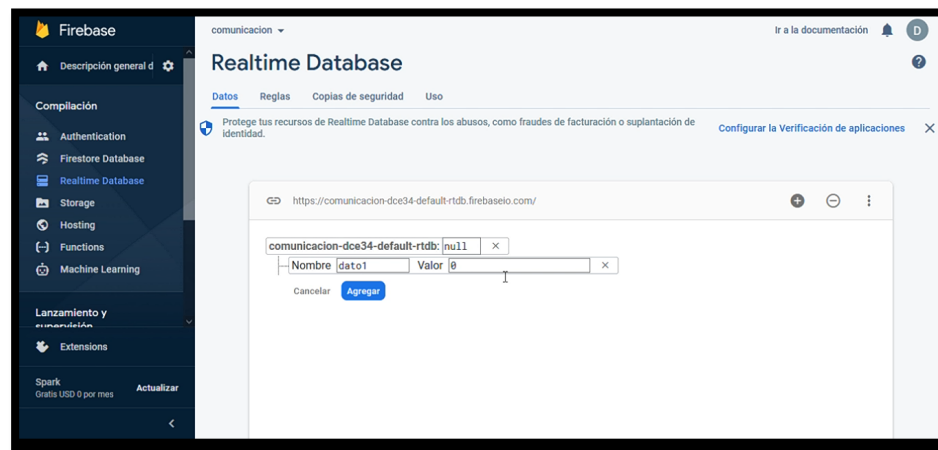
Figura 3. 19. Creación de Servicio Realtime



Elaborado por: Luis Montalván, Fernando Quinte

- En la figura 3.20 Creamos un lector de datos asignándole un nombre y un valor de cero, este valor de cero solo es como referencia ya que en esta parte se visualizaran los datos que se envían desde el sensor Emotiv Insight.

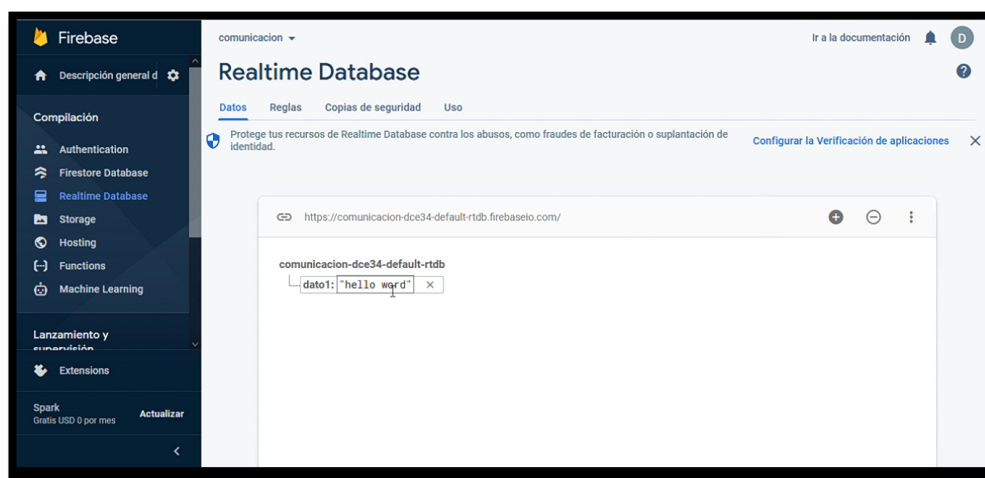
Figura 3. 20. Creación de lector de datos



Elaborado por: Luis Montalván, Fernando Quinte

- Una vez ya configurado el envío de datos en Node-Red solo resta visualizar los datos en este caso un dato de prueba verificación la comunicación como muestra la figura 3.21.

Figura 3. 21. Visualización de datos



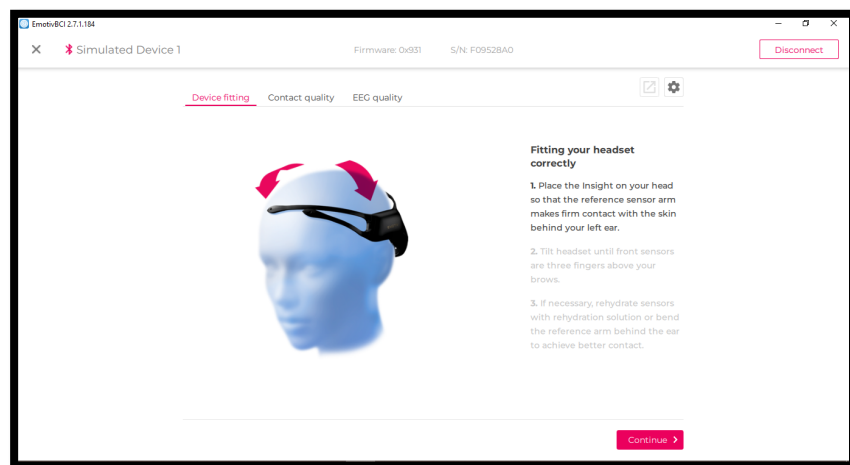
Elaborado por: Luis Montalván, Fernando Quinte

3.6. Entrenamiento del Sensor Emotiv Insight

En este apartado se indica el entrenamiento necesario para el uso del sensor Emotiv Insight para que los comandos mentales sean receptados de una manera óptima y personalizada para el usuario.

- Se empleará el software proporcionado por Emotiv.Inc llamado Emotiv BCI el cual nos proporciona instrucciones para una posición adecuada del sensor en la cabeza del usuario como se indica en la figura 3.22.

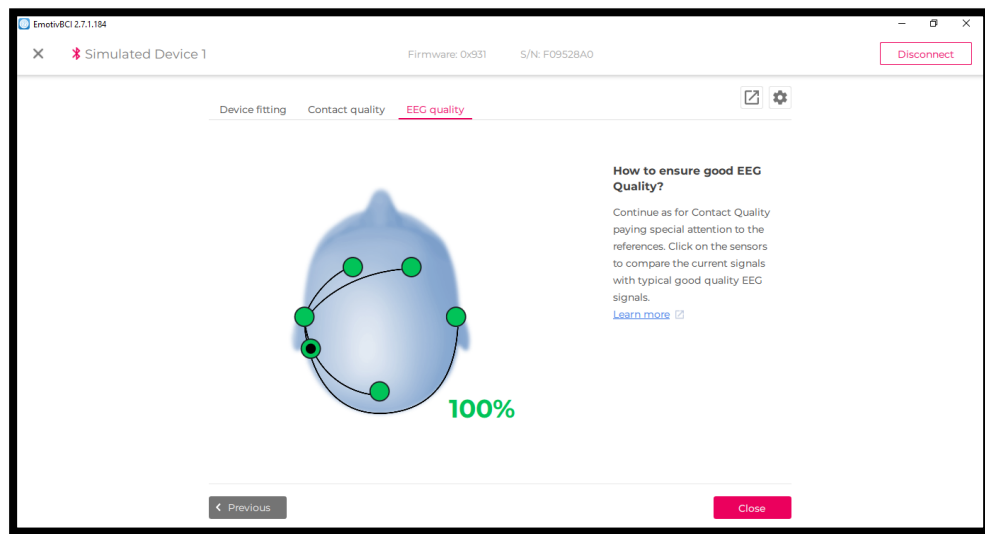
Figura 3. 22 Software Emotiv BCI



Elaborado por: Luis Montalván, Fernando Quinte

- Como se indica en la figura 3.23 verificaremos el contacto de los seis sensores con el cuero cabelludo donde el porcentaje de recepción deberá ser del 100% de la calidad de datos obtenidos de los sensores el encefalograma (EEG).

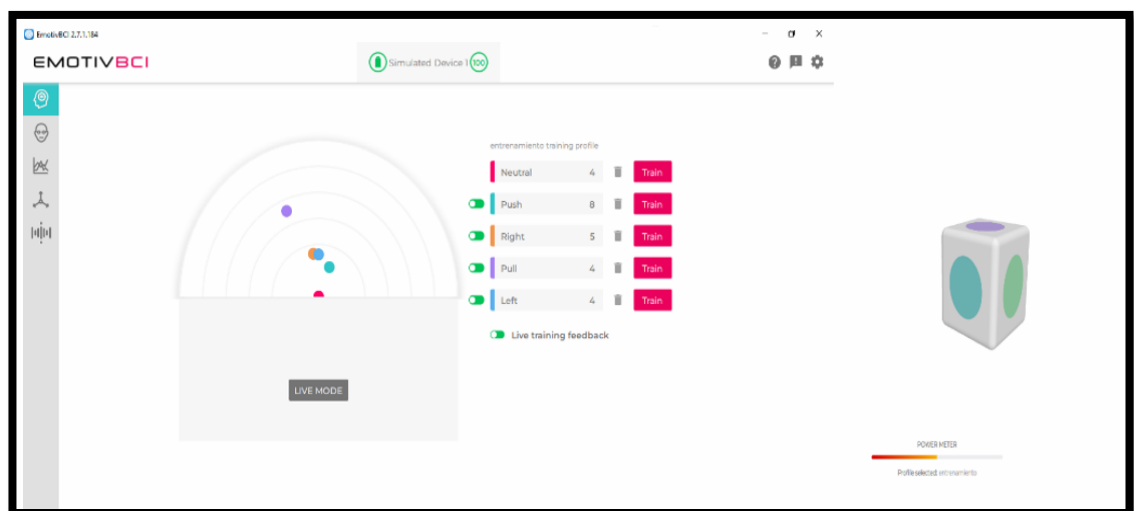
Figura 3. 23. Verificación de contacto de los sensores



Elaborado por: Luis Montalván, Fernando Quinte

- En la figura 3.24 se realiza el entrenamiento de los comandos mentales Push, Pull, Right, Left y Neutral tomando este punto de partida del entrenamiento, todos estos datos se almacenaran en una base de datos para su posterior uso.

Figura 3. 24 Entrenamiento de los comandos mentales



Elaborado por: Luis Montalván, Fernando Quinte

3.8. Implementación Física

El resultado final es la implementación física y sus respectivas pruebas en los diferentes comandos mentales, en este caso se visualizará como el usuario empieza por usar el comando mental Neutral para poder tomarlo como punto de partida y luego procede a realizar la prueba del comando Push que está conectado a una lámpara incandescente como se indica en la figura 3.25, el dispositivo ON-OFF podrá encenderse o apagarse usando el mismo comando mental gracias al algoritmo creado en el IDE de arduino especificado en la figura 3.12.

Figura 3. 25. Implementación física del comando Push



Elaborado por: Luis Montalván, Fernando Quinte

Cuando encendemos una salida con los diferentes comandos el usuario puede pasar a un estado denominado Neutral donde el usuario puede relajarse y sin temor a afectar el correcto funcionamiento del dispositivo mostrado en la figura 3.26.

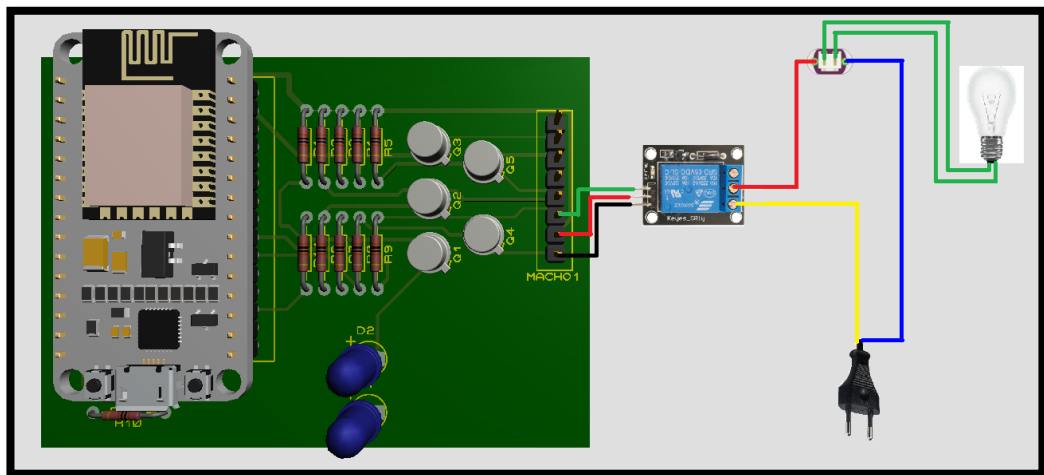
Figura 3. 26 Ilustración de estado Neutral



Elaborado por: Luis Montalván, Fernando Quinte

3.7. Diagrama Esquemático

Figura 3.27. Conexión de un canal de control



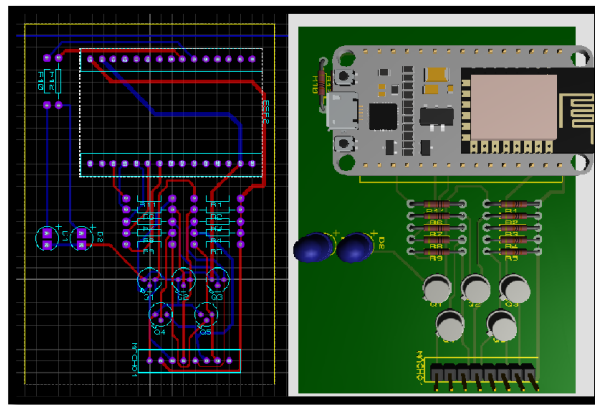
Fuente: Luis Montalván, Fernando Quinte

La figura 3.27 muestra las conexiones entre la placa PCB de control con el módulo ESP32, el modulo relé, tomacorriente y su alimentación a 110V, este esquema consta con dos diodos led que toman la función de indicadores, cada led se enciende de acuerdo a un caso en específico: El led rojo indica cuando el sistema está realizando

el encendido y conexiones respectivas mediante protocolo MQTT y el led verde indica que el sistema está listo para su uso

Con el objetivo de realizar el control del módulo relé se desarrolló una placa PCB de 8 x 5 centímetros como se indica en la figura.3.28 en la cual se tiene sobrepuesto el módulo ESP32 la cual emite las señales de control proporcionadas por el servidor en la nube que son previamente suministradas por el sensor Emotiv Insight, en la parte derecha podremos observar una representación 3D de la placa impresa con todos sus componentes.

Figura 3. 28 Diagrama Esquemático, Diseño Proteus

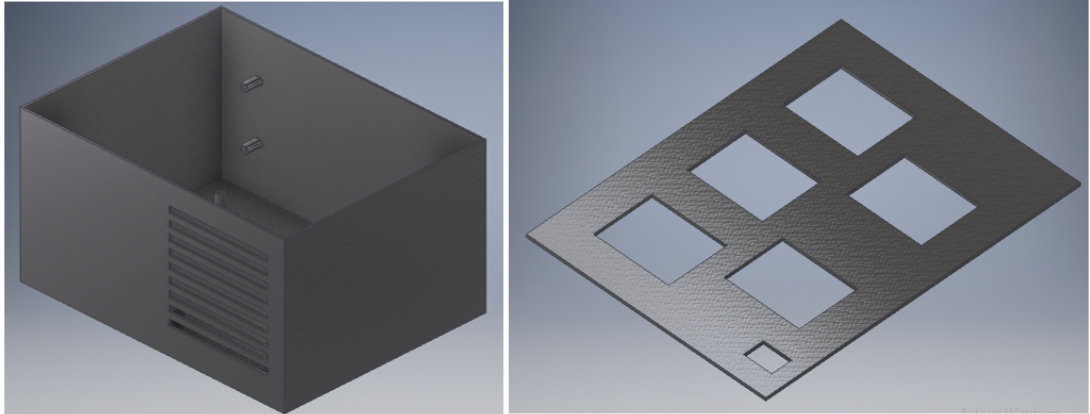


Elaborado por: Luis Montalván, Fernando Quinte

Para finalizar con el dispositivo se diseñó una estructura como se muestra en la figura 3.29 en la cual constará toda la conexión esquemática que consta de la tarjeta ordenador Raspberry, la placa PCB de control, Modulo Relé y los 5 tomacorrientes,

4 canales de control ON-OFF y 1 canal libre, además de rejillas de ventilación para evitar el sobrecalentamiento de los componentes.

Figura 3. 29 Planos Smartswitch



Elaborado por: Luis Montalván, Fernando Quinte

CAPÍTULO 4

4. PRUEBAS

4.1. Generalidades

Con el fin de realizar de pruebas de conexión y verificar los tiempos de respuesta entre cliente y servidor en la nube, se ha realizado comparaciones entre el tiempo de envío de datos hacia la nube donde este será registrado y el tiempo de recepción en el módulo ESP32 donde activará la salida correspondiente al comando mental asignado como se muestra en la figura 4.1.

Figura 4. 1. Tiempo de respuesta en cliente

```
22:25:55.800 -> Message arrived on topic: esp32/output1. Message: /com/neutral  
22:25:55.834 -> Changing output to  
22:25:55.834 -> NEUTRAL  
22:25:55.834 -> Message arrived on topic: esp32/output1. Message: neutral  
22:25:55.834 -> Changing output to
```

Elaborado por: Luis Montalván, Fernando Quinte

Figura 4. 2. Tiempo de respuesta en el servidor

Tiempo	Valor
24/6/2021 22:25:55	/com/neutral

Elaborado por: Luis Montalván, Fernando Quinte

Para lograr tener un valor tentativo de respuesta, se considera la ecuación 4.1 que compara el tiempo entre el dato del cliente y el dato del servidor en la nube

$$Tr = Tc - Ts \quad \text{Ec. (4.1)}$$

Tr = Tiempo de Respuesta total.

Ts = Tiempo de respuesta en el servidor en la nube.

Tc = Tiempo de respuesta en el cliente.

$Ts=22:25:55,000$; $Tc=22:25:55,834$

$Tr = 22:25:55,834 - 22:25:55,000$

$Tr = 00:00:00,834s$

El tiempo de respuesta entre cliente y servidor del comando mental Neutral en este dato tomado como ejemplo nos da un resultado de 843ms lo cual es un tiempo que se usará para dar un tiempo óptimo para un mantenimiento preventivo y correctivo.

4.2. Tablas de Tiempo de respuestas

Como se muestra en la tabla 4.1 para el comando Push se realizó el envío de 30 datos desde el sensor Emotiv Insight como indica el Anexo 3.1 teniendo como resultado un 90% de aciertos, un 10% de desaciertos y un tiempo de respuesta entre clientes y servidor de 1,834 segundos.

Tabla 4. 1. Resultados del envío de datos para el comando Push

Comando Push	
Comandos enviados	30
Aciertos	27
No aciertos	3
comandos ejecutados	27
Tiempo de respuesta (s)	0:00:01,843

Elaborado por: Luis Montalván, Fernando Quinte

En la tabla 4.2 para el comando Pull se realizó el envío de 30 datos desde el sensor Emotiv Insight como lo indica el Anexo 3.2 del cual se obtiene resultados de un 86.67% de aciertos, un 13.3% de desaciertos y un tiempo de respuesta entre cliente y servidor de 1,888 segundos.

Tabla 4. 2. Resultados del envío de datos para el comando Pull

Comando Pull	
Comandos enviados	30
Aciertos	26
No aciertos	4
comandos ejecutados	26
Tiempo de respuesta (s)	0:00:01,888

Elaborado por: Luis Montalván, Fernando Quinte

En la tabla 4.3 para el comando Right se realizó el envío de 30 datos desde el sensor Emotiv Insight como lo indica el Anexo 3.3 del cual se obtiene resultados de un 83.33% de aciertos, un 16.67% de desaciertos y un tiempo de respuesta entre cliente y servidor de 2,036 segundos.

Tabla 4. 3. Resultados del envío de datos para el comando Right

Comando Right	
Comandos enviados	30
Aciertos	25
No aciertos	5
comandos ejecutados	25
Tiempo de respuesta (s)	0:00:02,036

Elaborado por: Luis Montalván, Fernando Quinte

En la tabla 4.4 para el comando Left se realizó el envío de 30 datos desde el sensor Emotiv Insight como lo indica el Anexo 3.4 del cual se obtiene resultados de un 80% de aciertos, un 20% de desaciertos y un tiempo de respuesta entre cliente y servidor de 1,964 segundos.

Tabla 4. 4. Resultados del envío de datos para el comando Left

Comando Left	
Comandos enviados	30
Aciertos	24
No aciertos	6
comandos ejecutados	24
Tiempo de respuesta (s)	0:00:01,964

Elaborado por: Luis Montalván, Fernando Quinte

En la tabla 4.5 para el comando Neutral se realizó el envío de 30 datos desde el sensor Emotiv Insight como lo indica el Anexo 3.5 del cual se obtiene resultados de un 100% de aciertos, un 0% de desaciertos y un tiempo de respuesta entre cliente y servidor de 1,798 segundos

Tabla 4. 5. Resultados del envío de datos para el comando Neutral

Comando Neutral	
Comandos enviados	30
Aciertos	30
No aciertos	0
comandos ejecutados	30
Tiempo de respuesta (s)	0:00:01,798

Elaborado por: Luis Montalván, Fernando Quinte

Para obtener un rango de respuesta óptimo entre el cliente y servidor se hará un promedio de los datos obtenidos en los tiempos de respuestas para cada uno de los comandos mentales.

$$T_o = \frac{\sum Tr}{nTr} \quad \text{Ec. (4.2)}$$

T_o = tiempo óptimo

Tr = tiempo de respuesta

$$T_o = \frac{1.834 + 1.888 + 2.036 + 1.964 + 1.798}{5}$$

$$T_o = 1.904s \approx 2.5s$$

El tiempo óptimo resultó de 1.904 segundos lo cual se redondeará a un tiempo 2.5 segundos.

El criterio para definir el tiempo óptimo está basado en la media aritmética entre los tiempos de respuesta de los cinco comandos mentales, como se observa en la tabla 4.3 el tiempo de respuesta del comando right es mayor al tiempo óptimo, por lo tanto para no tener dificultades con este comando mental, se incrementó el rango de un tiempo óptimo a trabajar.

En la adquisición de datos para todos los comandos mentales arrojo porcentajes bajos de desaciertos, estos desaciertos son suscitados por varios factores tales como: una falta de concentración al momento de trabajar bajo un comando mental, los sensores de la diadema no estén haciendo un contacto optimo con el cuero cabelludo ya sea por abundante cabello o falta de lubricación en los sensores y también recalcando el estado anímico de la persona.

CAPÍTULO 5

CONCLUSIONES

La información proporcionada por el software Emotiv BCI nos ayuda a interpretar las señales eléctricas del cerebro obtenido desde el sensor Emotiv Insight para generar comandos mentales y usarlos como señales de control On-Off.

La información de los distintos dispositivos proporcionada para la culminación del proyecto determinó características importantes de cada uno, el sensor Emotiv Insight consta de 6 electrodos que captan las señales eléctricas para convertirlas en comandos mentales, la tarjeta ordenador Raspberry Pi 4B cumple la función de broker para el envío de datos, el módulo ESP32 realiza el algoritmo de control y el módulo relé se activó de acuerdo con las señales emitidas por los comando mentales, para que el envío de datos hacia la nube sea correcto.

Emotiv BCI permite la interpretación rápida de las señales detectadas previamente entrenadas, además de una licencia para compartir datos en aplicaciones de terceros en nuestro caso la tarjeta ordenador Raspberry Pi 4B ; por su parte la elección de Node-Red para el desarrollo de la adquisición de datos , interpretación, envío de datos a los servidores en la nube e interfaz gráfica.

El tiempo óptimo de respuesta entre el cliente y servidor dio como resultado 1.904 segundos como se muestra en el resultado de la Ec 4.2 y debido a que los datos obtenidos por el comando mental Right superan los 2 segundos de tiempo de respuesta, el rango establecido como tiempo de respuesta óptimo será 2.5 segundos.

Gracias al rango establecido para el tiempo de respuesta óptimo entre cliente y servidor podemos deducir que, si el tiempo de respuesta sobrepasa el rango admitido representa

un fallo en la comunicación del sistema ya sea en el envío de datos hacia el servidor en la nube o la recepción de datos en el módulo ESP32.

Para este proyecto se utilizó dos proveedores de servidores en la nube IBM Cloud de IBM Inc. y Firebase de Google, se decidió usar el servidor de IBM para realizar pruebas de tiempo de respuesta entre cliente y servidor, ya que este servidor nos proporciona datos en tiempo real en horas, minutos, segundos y milisegundos, en FireBase se muestra el dato en tiempo real pero no proporciona información sobre el tiempo de llegada de los datos en horas, minutos, segundos y milisegundos.

El sensor puede llegar a tener errores en la interpretación de datos por comandos mentales o gestos debido a que se adicionan un mayor número de acciones al momento de realizar el entrenamiento llegando a producir confusiones en las acciones a realizar, la solución que se usó fue modificar la sensibilidad del sensor desde el software Emotiv BCI. En los casos en los que se ajustó la sensibilidad se lo hizo con un margen de 0.5 para cada acción.

Algunos factores que perturban la interpretación de la señales son problemas de concentración, estados como cansancio, ruido del entorno, emociones del usuario; el cabello largo presenta un obstáculo entre el cuello cabelludo y los sensores complicando así la lectura de las señales cerebrales.

RECOMENDACIONES

Previo a la utilización del sensor Emotiv Insight se debe lubricar los electrodos con glicerina o una solución salina para lograr una mejor lectura de las señales cerebrales.

Dado que el cabello en largo o en abundancia puede alterar la detección de las señales cerebrales por parte de los sensores, se recomienda que la persona a usar el sensor Emotiv Insight tenga el cabello corto para no dificultar el contacto entre el cuero cabelludo y los electros del sensor Emotiv Insight.

Para evitar fallos en los protocolos de comunicación, Node-Red se recomienda utilizar a la tarjeta ordenador Raspberry Pi 4B como Broker ya que esta dispone de una mejor capacidad de memoria RAM y procesador.

BIBLIOGRAFÍA

World Health Organization. (2013). Informe mundial sobre la discapacidad. (O. M. Salud, Ed.).

Discapacidades, C. N. (11 de marzo de 2021). Consejo de discapacidades. Obtenido (CONADIS) de <https://www.consejodiscapacidades.gob.ec/#>.

Palaniappan, R. (2014). *Electroencephalogram-based Brain-Computer Interface*: En E. R. Miranda, Guide to Brain-Computer Music Interfacing (págs. 29-41). Londres, London: Springer-Verlag London.

Ramadan, R. A., Vasilakos, A. V. (Octubre de 2016). *Brain Computer Interface: Control Signals. Neurocomputing*, 223, 31-34. doi:10.1016/j.neucom.2016.10.024.

F. Ramos-Arguelles, G. Morales, S. Egozcue, R.M.Pabón, & M.T. Alonso, Navar (2009). *Técnicas básicas de electroencefalograma: principio y aplicaciones clínicas*, Pag: 2.

Villegas, M. B, & Rojas, F. M. (Septiembre de 2018). *Interfaz cerebro ordenador BCI mediante el uso de Emotiv Insight*.p:19, Departamento de Ciencias Exactas e Ingeniería, Universidad Católica Boliviana.

McGrath, M. (2016). Raspberry Pi 3 In Easy Steps (p 192). *Place of publication not identified: In Easy Steps Limited*

Loachamín Sangoquiza, R. R. (Enero 2021). *Análisis de una red inalámbrica mallada autoconfigurable, utilizando el módulo nodemcu ESP32 con el estándar 802.11*. Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional. Pag:19.

Jamal, I., (Septiembre 2020). *Sistema de Adquisición de Datos con ESP32*. Facultad de Ingeniería Electrónica Industrial y Automática, Universidad Politécnica de Catalunya.

Vistrónica, (2020), *Módulo Relé, Rescatado de:* <https://www.vistronica.com/potencia/modulo-rele-de-1-canal-detail.html>.

Murazzo, M, Millán, F, Rodriguez, N, Segura, D, Villafañe, D., (2012). *Desarrollo de aplicaciones para Cloud Computing*. Facultad de Ciencias Exactas, Físicas y Naturales, Universidad de San Juan. Pag:2

IBM Cloud, (19 de mayo del 2021). *Qué es la plataforma IBM Cloud*. Rescatado de : <https://cloud.ibm.com/docs/overview?topic=overview-what-is-platform&locale=es>.

Goñi, J. M. (Junio 2019). *EkitApp: conectando alumnos en una red social con Flutter y FireBase*, Facultad de Informatica, Universidad el País Vasco.

FireBase. (2020). *FireBase por plataforma*. Rescatado de: <https://FireBase.google.com/docs>.

Sicari, S, Rizzardi, A, Coen-Porisini, A, (13 de Diciembre de 2018). *Smart Transport and Logistics: A Node-Red implementation*. Italy, Departament of theorical and Applied Sciences, Varese University of Insubria.

Moreno Cerdá, F. (Enero de 2018). *Arquitectura de publicacion/subscripcion con protocolo MQTT*. Barcelona, Faculta de Ingeniería Técnica en Telecomunicaciones, Universidad Politénica de Catalunya.

E. Blanco, A. Francisco, & R. Edgar (2015). *Desarrollo de un Sistema de detección de Movimiento basado en flujo óptimo en Raspberry Pi*. Vol: 4, México, Asociación Mexicana de Mecatrónica.

ANEXOS

ANEXO 1. Código de Programación en Arduino.

Anexo 1.1 Código de inicialización de librerías y variables.

```
//incluir las librerías del wifi y PubSubClient para el modulo esp32
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>

//nombre y contraseña de la red wifi
const char* ssid = "CELERITY_MONTALVAN";
const char* password = "1722510060B.";

//la Ip del broker del protocolo mqtt
const char* mqtt_server = "192.168.1.11";

//especificamos el tipo de conexion que vamos
//a usar en este caso clientes
WiFiClient espClient;
PubSubClient client(espClient);

//variables para reconexion a protocolo mqtt
long lastMsg = 0;
char msg[50];

//variables para el encendido y apagado
int value1 = 0;
int value2 = 0;
int value3 = 0;
int value4 = 0;

// LED Pin
const int ledPin1 = 15;
const int ledPin2 = 2;
const int ledPin3 = 4;
const int ledPin4 = 16;
const int ledPin5 = 17;
const int ledPin6 = 5;
const int ledPin7 = 18;
```

Anexo 1.2 Inicialización de Pines como Output.

```
void setup() {
  //velocidad de carga
  Serial.begin(115200);
  //configuracion wifi
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  //pines como salidas
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin7, OUTPUT);
  //indicadores led's del encendido
  digitalWrite(ledPin1, HIGH);
  delay(1000);
  digitalWrite(ledPin6, HIGH);
  delay(60000);
  digitalWrite(ledPin6, LOW);
  digitalWrite(ledPin7, HIGH);
}
```

Anexo 1.3 Conexión a la Red Wifi

```
//conexion a la red wifi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  //reconexion a la red wifi
  while (WiFi.status() != WL_CONNECTED)
    delay(500);
    Serial.print(".");
  }
  //estado de conexion e ip adquirida
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```


Anexo 1.4 Transformación de tópicos a mensaje temporal.

```
//trasformacion de topic a mensaje temporal
void callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    String messageTemp;

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }

    Serial.println();
}
```

Anexo 1.5 “IF” de comandos mentales.

```
// si el mensaje es correcto entra o los diferentes casos If's
if (String(topic) == "esp32/output1") {
    Serial.print("Changing output to ");
    Serial.println("");
    if (messageTemp == "/com/neutral") {
        Serial.println("NEUTRAL");
        client.publish("esp32/output1", "neutral");
    }
}

if (String(topic) == "esp32/output2") {
    Serial.print("Changing output to ");
    Serial.println("");
    if (messageTemp == "/com/push" && value1 == 0) {
        Serial.println("push ON");
        client.publish("esp32/output2", "1");
        digitalWrite(ledPin2, HIGH);
        value1++;
    }
    else if (messageTemp == "/com/push" && value1 == 1) {
        Serial.println("push OFF");
        client.publish("esp32/output2", "0");
        digitalWrite(ledPin2, LOW);
        value1 = 0;
    }
}
}
```

Anexo 1.6 Conexión de Protocolo MQTT

```
// conexion al protocolo mqtt
void reconnect() {

    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

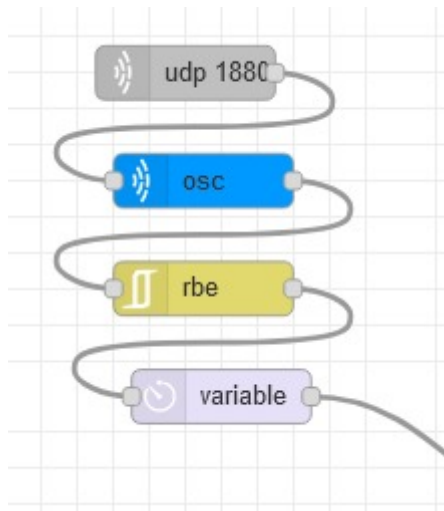
        if (client.connect("ESP8266Client")) {
            Serial.println("connected");
            // Subscriptores
            client.subscribe("esp32/output1");
            client.subscribe("esp32/output2");
            client.subscribe("esp32/output3");
            client.subscribe("esp32/output4");
            client.subscribe("esp32/output5");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            //esperar 5 segundos para la reconexion al fallar
            delay(5000);
        }
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}

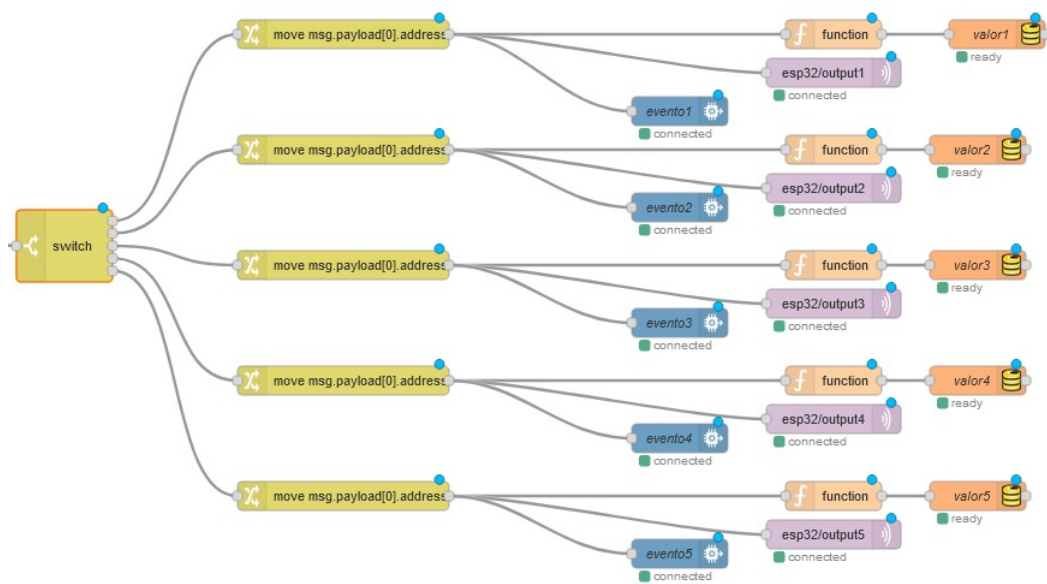
//tiempo para reconexion al fallar
long now = millis();
if (now - lastMsg > 5000) {
```

ANEXO 2. Programación Node-Red.

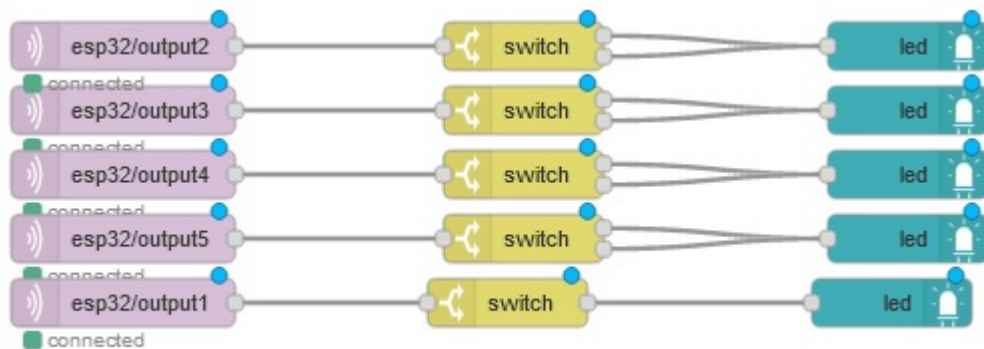
Anexo 2.1 Protocolo de licencia (UDP)



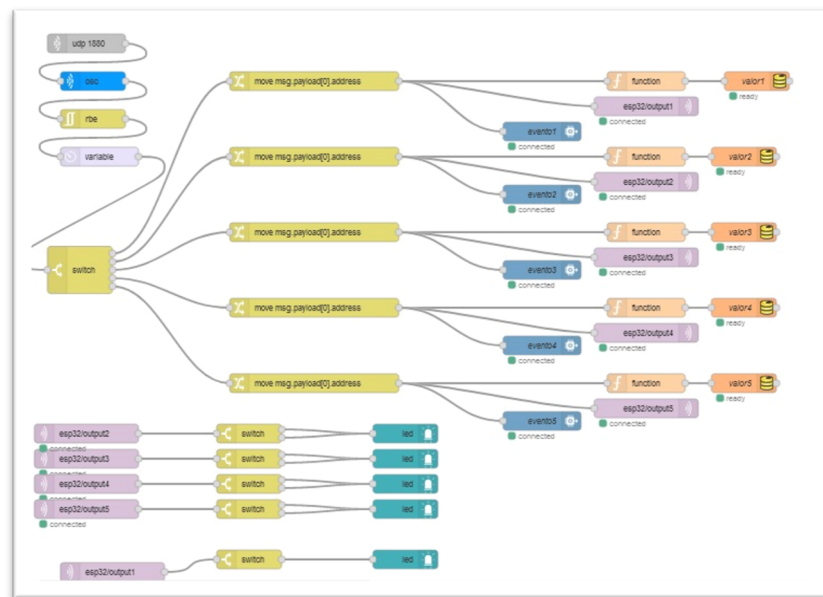
Anexo 2.2 Procesamiento de Datos a los servidores y ESP32



Anexo 2.3 Dashboard



Anexo 2.4 Programación completa



ANEXO 3. Tablas de Datos de comandos mentales.

Anexo 3.1 Tabla de Datos comando Push

	COMNADO PUSH			
NUMERO DE PRUEBA	¿EJECUTO EL COMANDO?		HORA LOCAL (h:m:s:ms)	HORA SERVIDOR (h:m:s:ms)
	SI	NO		
1	X		22:25:55,834	22:25:57,334
2		X	x	x
3	X		22:26:04,152	22:26:05,702
4	X		22:26:09,742	22:26:12,042
5	X		22:26:16,663	22:26:18,463
6	X		22:26:22,486	22:26:24,437
7	X		22:26:28,458	22:26:29,958
8	X		22:26:33,520	22:26:35,520
9	X		22:26:39,854	22:26:41,404
10	X		22:26:45,987	22:26:48,287
11		X	x	x
12	X		22:26:58,385	22:27:00,336
13	X		22:27:04,000	22:27:05,500
14	X		22:27:09,471	22:27:11,471
15	X		22:27:15,236	22:27:16,786
16	X		22:27:20,963	22:27:23,263
17	X		22:27:26,753	22:27:28,553
18	X		22:27:32,357	22:27:34,308
19	X		22:27:38,582	22:27:40,082
20	X		22:27:44,367	22:27:46,367
21	X		22:27:50,674	22:27:52,224
22	X		22:27:56,874	22:27:59,174
23	X		22:28:03,746	22:28:05,546
24		X	x	x
25	X		22:28:15,471	22:28:16,971
26	X		22:28:20,563	22:28:22,563
27	X		22:28:26,985	22:28:28,535
28	X		22:28:32,563	22:28:34,863
29	X		22:28:38,968	22:28:40,768
30	X		22:28:44,593	22:28:46,544

Anexo 3.2 Tabla de Datos comando Pull

	COMNADO PULL			
NUMERO DE PRUEBA	¿EJECUTO EL COMANDO?		HORA LOCAL (h:m:s:ms)	HORA SERVIDOR (h:m:s:ms)
	SI	NO		
1	X		22:28:50,950	22:28:52,496
2	X		22:28:54,985	22:28:57,525
3	X		22:29:00,541	22:29:02,541
4		X	X	X
5	X		22:29:12,637	22:29:14,267
6	X		22:29:20,452	22:29:22,302
7	X		22:29:26,267	22:29:27,814
8	X		22:29:31,569	22:29:33,569
9	X		22:29:38,678	22:29:40,224
10	X		22:29:44,789	22:29:47,329
11	X		22:29:51,741	22:29:53,741
12	X		22:29:58,634	22:30:00,374
13		X	X	X
14	X		22:30:10,796	22:30:12,646
15	X		22:30:16,540	22:30:18,087
16	X		22:30:22,596	22:30:24,596
17	X		22:30:28,965	22:30:30,511
18	X		22:30:34,746	22:30:37,286
19	X		22:30:40,967	22:30:42,967
20	X		22:30:46,789	22:30:48,529
21	X		22:30:52,693	22:30:54,323
22	X		22:31:00,874	22:31:02,724
23	X		22:31:07,637	22:31:09,184
24	X		22:31:13,978	22:31:15,978
25		X	X	X
26	X		22:31:25,784	22:31:28,324
27	X		22:31:31,746	22:31:33,746
28	X		22:31:38,687	22:31:40,427
29	X		22:31:44,645	22:31:46,275
30		X	X	X

Anexo 3.3 Tabla de Datos comando Right

	COMNADO RIGHT			
NUMERO DE PRUEBA	¿EJECUTO EL COMANDO?		HORA LOCAL (h:m:s:ms)	HORA SERVIDOR (h:m:s:ms)
	SI	NO		
1	X		22:31:55,800	22:31:57,300
2	X		22:32:01,540	22:32:03,124
3	X		22:32:08,547	22:32:10,147
4	X		22:32:14,574	22:32:16,574
5		X	X	X
6	X		22:32:26,874	22:32:29,014
7	X		22:32:32,698	22:32:34,438
8	X		22:32:38,748	22:32:41,749
9	X		22:32:44,768	22:32:47,218
10	X		22:32:50,748	22:32:52,248
11	X		22:32:57,321	22:32:58,905
12	X		22:33:03,474	22:33:05,074
13		X	X	X
14	X		22:33:14,744	22:33:17,394
15	X		22:33:20,741	22:33:22,881
16	X		22:33:26,896	22:33:28,636
17	X		22:33:33,985	22:33:36,986
18	X		22:33:39,847	22:33:42,297
19		X	X	X
20	X		22:33:50,748	22:33:52,332
21	X		22:33:58,698	22:34:00,298
22	X		22:34:04,987	22:34:06,987
23	X		22:34:11,687	22:34:14,337
24		X	X	X
25	X		22:34:23,748	22:34:25,488
26	X		22:34:29,748	22:34:32,749
27	X		22:34:35,784	22:34:38,234
28		X	X	X
29	X		22:34:48,746	22:34:50,330
30	X		22:34:54,988	22:34:56,588

Anexo 3.4 Tabla de Datos comando Left

	COMNADO LEFT			
NUMERO DE PRUEBA	¿EJECUTO EL COMANDO?		HORA LOCAL (h:m:s:ms)	HORA SERVIDOR (h:m:s:ms)
	SI	NO		
1	X		22:35:00,520	22:35:02,220
2	X		22:35:06,547	22:35:09,247
3		X	X	X
4	X		22:35:18,749	22:35:21,049
5	X		22:35:25,599	22:35:28,249
6	X		22:35:31,742	22:35:33,482
7	X		22:35:37,748	22:35:39,898
8	X		22:35:43,471	22:35:45,371
9	X		22:35:49,784	22:35:50,284
10		X	X	X
11	X		22:36:02,766	22:36:04,466
12	X		22:36:04,750	22:36:07,450
13	X		22:36:10,746	22:36:12,046
14	X		22:36:16,476	22:36:18,776
15	X		22:36:22,947	22:36:25,597
16		X	X	X
17	X		22:36:35,680	22:36:37,830
18	X		22:36:41,658	22:36:43,558
19	X		22:36:47,320	22:36:47,820
20	X		22:36:52,840	22:36:54,940
21		X	X	X
22	X		22:37:04,766	22:37:07,466
23	X		22:37:11,747	22:37:13,047
24	X		22:37:17,699	22:37:19,999
25		X	X	X
26	X		22:37:29,570	22:37:31,310
27	X		22:37:35,800	22:37:37,950
28	X		22:37:41,001	22:37:42,901
29		X	X	X
30	X		22:37:53,900	22:37:56,000

Anexo 3.5 Tabla de Datos comando Neutral

	COMNADO NEUTRAL			
NUMERO DE PRUEBA	¿EJECUTO EL COMANDO?		HORA LOCAL (h:m:s:ms)	HORA SERVIDOR (h:m:s:ms)
	SI	NO		
1	X		22:38:00,500	22:38:02,000
2	X		22:38:06,700	22:38:08,700
3	X		22:38:12,540	22:38:13,890
4	X		22:38:18,740	22:38:20,340
5	X		22:38:25,700	22:38:27,000
6	X		22:38:31,820	22:38:33,820
7	X		22:38:37,985	22:38:39,985
8	X		22:38:43,710	22:38:46,410
9	X		22:38:49,500	22:38:51,410
10	X		22:38:55,780	22:38:57,280
11	X		22:39:02,760	22:39:04,760
12	X		22:39:08,450	22:39:09,800
13	X		22:39:14,780	22:39:16,380
14	X		22:39:20,860	22:39:22,160
15	X		22:39:26,830	22:39:28,830
16	X		22:39:32,720	22:39:34,720
17	X		22:39:39,630	22:39:42,330
18	X		22:39:45,780	22:39:47,690
19	X		22:39:51,910	22:39:53,410
20	X		22:39:57,060	22:39:59,060
21	X		22:40:03,610	22:40:04,960
22	X		22:40:09,710	22:40:11,310
23	X		22:40:16,600	22:40:17,900
24	X		22:40:22,720	22:40:24,720
25	X		22:40:28,410	22:40:30,410
26	X		22:40:33,500	22:40:36,200
27	X		22:40:39,850	22:40:41,760
28	X		22:40:45,100	22:40:46,600
29	X		22:40:51,660	22:40:53,660
30	X		22:40:57,153	22:40:58,503